MASTER EN SCIENCES INFORMATIQUES À FINALITÉ SPÉCIALISÉE EN DATA SCIENCE

Conception d'un système de recommandation de littérature scientifique

Albert, Julien

Award date:
2020

Awarding institution:
Universite de Namur

Université de Namur
Faculté d'informatique
Année académique 2019–2020

Conception d'un système de recommandation de littérature scientifique

Julien ALBERT

Maître de stage : Mathieu GOEMINNE

Promoteur : _____ (Signature pour approbation du dépôt - REE art. 40)
Pr. Benoît FRENAY

Mémoire présenté en vue de l'obtention du grade de
Master en Sciences Informatiques.

Remerciements

Je tiens tout d'abord à remercier mon promoteur, Monsieur Benoit Frenay, Professeur à l'UNamur, pour sa disponibilité et ses judicieux conseils tout au long de la réalisation du mémoire.

Je désire aussi remercier mon maître de stage, Monsieur Mathieu Goeminne, Ingénieur de recherche au CETIC, pour sa disponibilité et la qualité de son suivi.

Je remercie également :

Madame Nathalie Grandjean et Messieurs Antoine Clarinval, Bruno Dumas et Pierre-Yves Schobbens pour m'avoir permis de les interviewer.

Mesdames Laurie Ceccotti et Bérengère Nihoul, et Messieurs Maher Badri, Simon Jacquet et Faiez Zalila pour leur participation à l'atelier d'analyse des besoins.

Enfin, je souhaite exprimer toute ma reconnaissance envers mes parents pour m'avoir toujours soutenu durant ces six dernières années.

Résumé/Abstract

L'objectif de ce travail est de concevoir un système de recommandation de littérature scientifique. L'originalité de ce système est de permettre à l'utilisateur de spécifier ses intérêts sous forme d'une sélection d'articles scientifiques et de multiplier les cycles de recherche grâce à une recommandation en temps réel. L'ambition est d'ailleurs de proposer une véritable alternative aux moteurs de recherche en littérature scientifique.

Outre la traditionnelle revue de littérature, ce travail débute par l'analyse des besoins des utilisateurs afin de définir les exigences fonctionnelles et non-fonctionnelles du système à concevoir. Une étude comparative est ensuite menée pour identifier les méthodes de recommandation les plus appropriées. La conception proprement dite du système peut alors être réalisée et concrétisée par l'implémentation d'un premier prototype. Enfin, ce prototype est évalué par le biais d'une étude utilisateur.

Mots-clés : système de recommandation, littérature scientifique, prototype, comparaison de méthodes de recommandation, étude utilisateur, protocole d'évaluation hors-ligne.

The objective of this work is to design a scientific paper recommender system. The originality of this system is to allow the user to specify his interests in the form of a selection of scientific articles and to multiply the search cycles thanks to a real-time recommendation. The ambition is to offer a real alternative to scientific paper search engines.

In addition to the traditional literature review, this work begins with a user needs analysis to define the functional and non-functional requirements of the system to be designed. A comparative study is then conducted to identify the most appropriate recommendation methods. The actual design of the system can then be carried out and concretized by the implementation of a first prototype. Finally, this prototype is evaluated through a user study.

Keywords : recommender system, scientific literature, prototype, comparison of recommendation methods, user study, off-line evaluation protocol.

Table des matières

Chapitre 1

Introduction

La production scientifique est devenue considérable aujourd'hui, avec plusieurs centaines de millions de références disponibles [Gusenbauer, 2019] et un taux de croissance annuel entre 2 et 3 % selon les estimations [Bornmann and Mutz, 2015]. La recherche bibliographique, qui essentielle pour la recherche et l'innovation, est donc devenue de plus en plus complexe. Des tâches comme la veille bibliographique ou la réalisation d'états de l'art pour des publications en cours de rédaction sont devenus des exercices ardus, notamment pour les jeunes chercheurs où lorsque le domaine concerné est peu familier.

Dans ce contexte, les outils de filtrage de l'information deviennent indispensables pour explorer de manière ciblée la littérature scientifique. Les moteurs de recherche sont sans doute la solution la plus fréquemment utilisée pour la recherche bibliographique [Khabsa et al., 2016]. Néanmoins, ils peuvent parfois ne pas être les plus adaptés, par exemple lorsque les bons mots-clés ne sont pas bien identifiés ou lorsque l'utilisateur souhaite introduire de la diversité dans les résultats obtenus.

Les systèmes de recommandation sont une alternative aux moteurs de recherche comme outils de filtrage de l'information. Ils sont utilisés dans de nombreux domaines et leur capacité à répondre efficacement au problème de la surcharge d'information n'est plus à démontrer [Ricci et al., 2015]. De plus, ils sont capables de capter finement les besoins des utilisateurs en croisant de multiples canaux, et ainsi fournir les résultats les plus adaptés. Ces éléments font que l'utilisation des systèmes de recommandation a toute sa pertinence dans le cadre de la recherche bibliographique [Le et al., 2019].

Ce travail a pour origine un proposition de stage du CETIC [1] sous la supervision de Mathieu Goeminne, ingénieur de recherche en sciences des données. Celle-ci consiste à développer un système de recommandation à destination des membres du CETIC, et *in fine*, à démontrer l'intérêt d'un outil de ce genre. Afin de préciser le contexte d'utilisation, le scénario ciblé est celui d'un chercheur en train de rédiger un article avec une bibliographie partielle et souhaitant, sur base de celle-ci, des recommandations bibliographiques. Enfin, ce mémoire est concrétisé par la réalisation d'un prototype fonctionnel sous forme d'une application web [2].

1. `https://www.cetic.be/Systeme-de-recommandation-pour-les-references-bibliographiques`
2. Accessible à l'adresse suivante : `https://stage-bibliographie.cetic.be/`

1.1 Objectifs et méthodologie

Le premier objectif, qui est également la première étape, consiste en la réalisation d'un état de l'art du domaine d'intérêt. Celui-ci a évidemment pour but de comprendre les différentes méthodes et approches de recommandation, ainsi que les spécificités propres au domaine de la littérature scientifique. L'idée principale est d'explorer le paysage de la recommandation en littérature scientifique et d'identifier les différents axes de recherche. Cette partie est l'objet du chapitre 2. Une attention particulière est également portée à la notion d'efficacité d'un système de recommandation entendue comme sa capacité à rencontrer les besoins des utilisateurs et autres parties prenantes. L'efficacité ainsi que les différents modes d'évaluation sont explorés dans le chapitre 3.

Il est ensuite nécessaire d'analyser en profondeur le scénario de recommandation ciblé. Cette analyse a pour but de définir les exigences fonctionnelles et non-fonctionnelles du système de recommandation. Celles-ci sont bien sûr principalement déterminées en concertation avec les différentes parties prenantes, mais également en prenant compte les restrictions liées aux sources de données disponibles, qui font aussi l'objet d'une analyse spécifique, en plus de l'analyse orientée utilisation. Cette partie est l'objet du chapitre 4.

L'étape suivante, développée dans le chapitre 5, consiste en la comparaison et la sélection d'une ou plusieurs méthodes de recommandation à partir des résultats d'une évaluation hors-ligne, c'est-à-dire ne nécessitant par l'intervention d'utilisateurs réels. Les méthodes à évaluer sont choisies sur base des enseignements tirés de l'état de l'art ainsi que des exigences fonctionnelles et non-fonctionnelles propres au scénario ciblé. Il s'agit donc de concevoir et de mettre en place un protocole robuste permettant l'évaluation comparative de méthodes fort différentes. Afin de répondre aux exigences, celui-ci prend également en compte différents aspects d'efficacité par le biais de mesures spécifiques.

Afin de concrétiser le travail réalisé, un premier prototype est conçu et implémenté. Il permet d'identifier une série de problèmes liés à la mise en production d'un système de recommandation de ce type et d'envisager des pistes de solution. Ces problèmes sont notamment liés à la gestion des données et au processus de recommandation. Enfin, il est également l'occasion de proposer une première interface utilisateur. Sa conception et sa réalisation sont décrites dans le chapitre 6.

L'objectif du prototype est également d'être une preuve de concept permettant de montrer l'intérêt du système conçu dans ce travail. Ce qui est réalisé par le biais d'une étude utilisateur s'intéressant à l'utilité générale du prototype et sa capacité à répondre aux différents besoins spécifiques des utilisateurs potentiels. Cette étude et les résultats obtenus sont décrits dans le chapitre 7.

Enfin, la conclusion (chapitre 8) est l'occasion de revenir sur le travail réalisé en mettant en avant ses forces et ses faiblesses, ainsi que les perspectives d'amélioration les plus intéressantes.

1.2 Présentation du domaine

Un système de recommandation est un outil de filtrage d'information qui propose à un **utilisateur** une sélection personnalisée d'**items** dans un corpus donné et correspondant aux buts et caractéristiques de celui-ci [de Gemmis et al., 2017]. Pour cela, il collecte des informations sur les préférences de l'utilisateur afin de le modéliser. Ces informations sont notamment liées aux **transactions** entre utilisateurs et items [Ricci et al., 2015]. Il peut s'agir d'informations données explicitement comme des évaluations ou implicitement comme des clics ou des téléchargements. Un

système de recommandation peut également prendre en compte le **contexte** de recommandation, comme la localisation de l'utilisateur ou la moment de la transaction.

Dans le cadre de la recommandation de littérature scientifique, les items considérés sont les publications scientifiques au sens large, c.à-d. les articles, les actes de conférences, les livres, etc., accessibles par le biais d'outils de recherche dédiés. Dans ce travail, le terme *article* est utilisé de manière générique pour regrouper les différents types de publications scientifiques. Les utilisateurs regroupent l'ensemble des personnes susceptibles de consulter ces documents. Leurs profils et leurs motivations sont assez variés. Et les transactions caractérisent les relations entre ces utilisateurs et les articles, par exemple la citation ou la consultation.

Enfin, il convient de distinguer la recommandation globale qui concerne le cas général défini ci-dessus et appliqué à la littérature scientifique, et la recommandation locale qui correspond à un cas plus spécifique. Ce dernier consiste à proposer à l'utilisateur à partir d'un extrait de texte, typiquement une partie d'article, une ou plusieurs suggestions de citations [Huang et al., 2015]. La recommandation locale n'est pas prise en compte dans ce travail.

La terminologie utilisée dans la suite de ce travail est définie par Beel et al. [2016]. Celui-ci considère un **système de recommandation** comme un logiciel de recommandation fonctionnel. Il comprend l'implémentation d'une ou plusieurs méthodes de recommandation ainsi que les différents composants nécessaires à sa mise en production (interface, données, etc.). Un **scénario de recommandation** décrit l'ensemble du cadre dans lequel est déployé un système de recommandation. Une **méthode de recommandation** est une méthode décrite de manière suffisamment précise pour permettre son implémentation. Cette description comprend notamment le fonctionnement général, l'architecture et les différents algorithmes utilisés (généralement en pseudo-code). À noter que Beel et al. [2016] utilise plutôt le terme d'algorithme mais il ne semble pas sémantiquement adapté à la réalité qu'il recouvre. Une **classe de recommandation** regroupe un ensemble de méthodes de recommandation basées sur une même idée générale, par exemple l'exploitation du contenu des items. Enfin, une **approche de recommandation** est un sous-ensemble d'une classe regroupant des méthodes appliquant de manière similaire l'idée générale de la classe pour générer des recommandations. Cependant, la description d'une approche n'est pas suffisamment précise pour permettre l'implémentation, ce qui la distingue d'une méthode de recommandation.

Afin d'envisager la question de l'évaluation, il faut tout d'abord définir le concept d'**efficacité** comme la capacité d'un système ou d'une méthode à atteindre ses objectifs [Beel et al., 2016]. Il peut s'agir d'objectifs fonctionnels comme répondre à différents types de besoins informationnelles et non-fonctionnels comme la scalabilité ou la réactivité du système. Et l'**évaluation** est une estimation de l'efficacité d'un système ou d'une méthode de recommandation [Beel et al., 2016].

Afin de décrire le processus de recommandation, le modèle proposé par Ricci [2017] pour les méthodes étudiées semble le plus adéquat (voir figure 1.1). Selon lui, le processus de recommandation peut être décomposé en trois tâches :

1. la **détermination des préférences de l'utilisateur**, c'est-à-dire la collecte d'évaluations d'items par des utilisateurs, éventuellement en prenant en compte le contexte,
2. la **prédiction des évaluations des items** à partir des données collectées,
3. et enfin la **génération des recommandations** à partir des évaluations prédites.

D'autres modèles existent. Par exemple, Isinkaye et al. [2015] considère que le processus de recommandation se déroule en trois phases : la collecte d'informations afin de construire un modèle de l'utilisateur, l'apprentissage pour filtrer et exploiter les informations collectées et la prédiction-

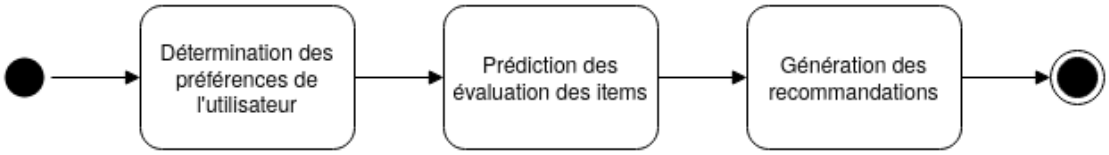

FIGURE 1.1 – Processus général de recommandation.

recommandation. Cependant, ce modèle a un pouvoir explicatif moindre notamment parce que la notion d'apprentissage n'est pas systématiquement présente dans les méthodes étudiées, et à cause de la vision monolithique de la phase de prédiction-recommandation. Khusro et al. [2016] propose également un modèle général du processus de recommandation mais celui-ci est trop orienté vers les méthodes basées sur les transactions entre utilisateurs et items, au détriment d'autres comme celles basées sur le contenu des items.

Chapitre 2

Revue de la littérature

De nombreuses revues de la littérature existent sur les systèmes de recommandation en général [Lu et al., 2015, Park et al., 2012] et spécifiques à la recommandation de littérature scientifique [Bai et al., 2019, Beel et al., 2016]. La présente revue de la littérature s'appuie largement sur ces travaux ainsi qu'un corpus comprenant une septantaine de méthodes de recommandation publiées en 2014 et après (voir annexe A.3). Elle tente tout de même de se démarquer en approfondissant certains aspects importants pour ce travail et en présentant certaines approches et méthodes récentes. Pour chacun des aspects développés, l'idée est de partir du champ général des systèmes de recommandation afin d'introduire les notions et enjeux principaux, et ensuite d'évoquer les spécificités de la recommandation de littérature scientifique.

2.1 Détermination des préférences de l'utilisateur

2.1.1 Définitions

La modélisation de l'utilisateur est centrale dans la capacité d'un système de recommandation à permettre des recommandations personnalisées [Ricci et al., 2015]. La première tâche d'un système de recommandation est d'ailleurs de déterminer les préférences des utilisateurs [Ricci, 2017]. Il s'agit concrètement de collecter un ensemble de transactions, c'est-à-dire des évaluations d'items par des utilisateurs. La collecte comprend également d'autres éléments jugés pertinents comme les attributs de l'utilisateur ou le contenu des items [Isinkaye et al., 2015]. Ricci [2017] définit formellement une transaction de la manière suivante. Soit $\mathcal{U}$, $\mathcal{I}$ et $\mathcal{R}$ respectivement les ensembles des utilisateurs, des items et des valeurs de l'échelle d'évaluation utilisée. La transaction entre un item i et un utilisateur u est définie comme le triplet $(u, i, r(u, i)) \in \mathcal{U} \times \mathcal{I} \times \mathcal{R}$ avec $r : \mathcal{U}, \mathcal{I} \rightarrow \mathcal{R}$ qui est la fonction d'évaluation. L'échelle d'évaluation utilisée peut être [Ricci et al., 2015] :

- numérique : intervalle de valeurs continues, par ex. $[0..5]$,
- ordinale : ensemble fini de valeurs discrètes ordonnées, par ex. $\{1, 2, 3, 4, 5\}$,
- binaire : évaluation positive ou négative, par ex. $\{-1, 1\}$,
- ou unaire : évaluation ou absence d'évaluation, par ex. $\{?, 1\}$.

2.1.2 Transactions

Transactions explicites

Les transactions peuvent être des évaluations explicites fournies par les utilisateurs et collectées par le système. Celles-ci offrent une bonne fiabilité étant donné qu'elles ne sont pas inférées d'un comportement tiers. Mais elles sont plutôt difficiles à obtenir en quantité suffisante car elles dépendent de la volonté des utilisateur [Isinkaye et al., 2015]. Dans le cas de la recommandation de littérature scientifique, le scénario le plus fréquemment identifié est celui-ci d'un utilisateur sélectionnant explicitement un ou plusieurs articles afin de se voir recommander des articles similaires (voir figure 2.1). L'échelle d'évaluation utilisée est alors unaire.

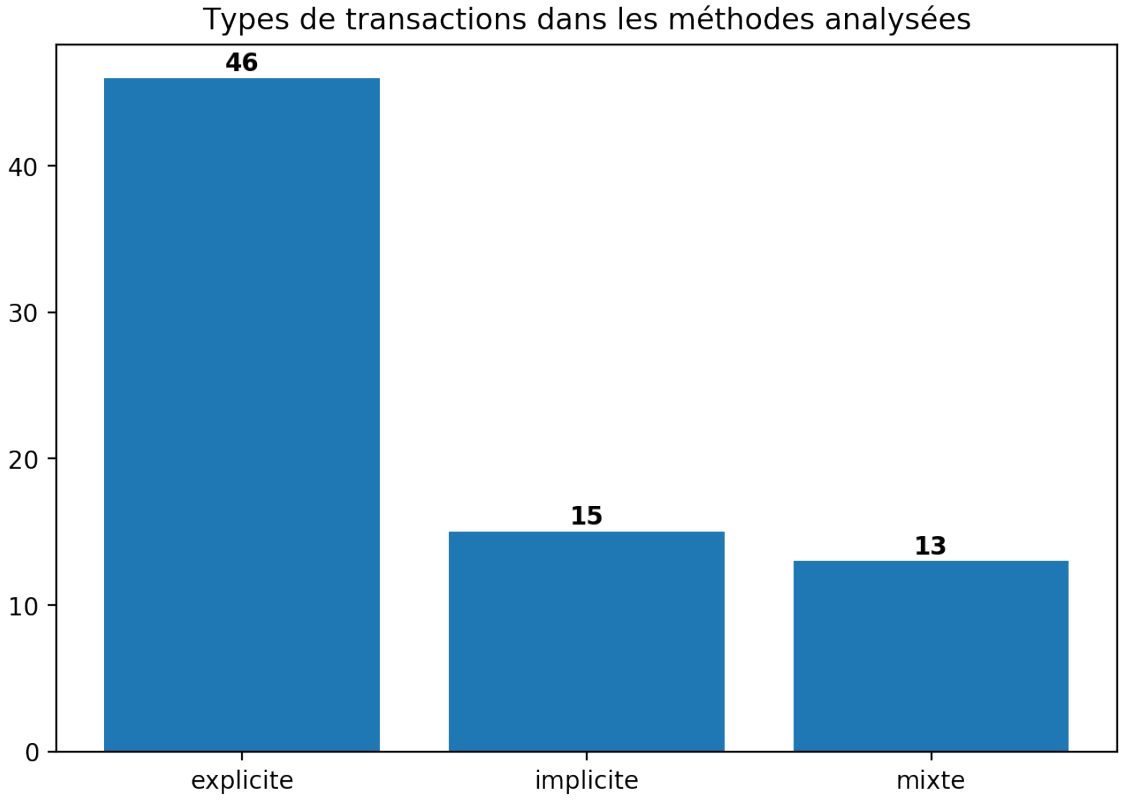

FIGURE 2.1 – Types de transactions employés dans les méthodes analysées (voir annexe A.3).

Dans le cas de la recommandation de littérature scientifique, les préférences de l'utilisateur peuvent également être inférées à partir d'une requête textuelle sous forme de mots-clés [Mu et al., 2018, Sesagiri Raamkumar et al., 2017] ou d'un texte court (par exemple un abstract ou la description d'un projet de recherche) [Jiang et al., 2015, Yang et al., 2019]. Avec cette approche, un système de recommandation n'est d'ailleurs pas fort différent d'un moteur de recherche où l'utilisateur soumet une requête exprimant son intérêt et reçoit en retour une liste de résultats [Beel et al., 2016]. Dans la plupart des cas, les méthodes de recommandation utilisant cette approche peuvent être adaptées à un scénario avec sélection d'articles d'intérêt simplement en utilisant les méta-données appropriées (par exemple le titre, l'abstract et/ou les mots-clés).

Transactions implicites

Afin de pallier la difficulté de collecter des évaluations explicites, des évaluations implicites peuvent être inférées à partir de l'observation du comportement des utilisateurs (i.e. des actions telles que des clics, des achats, la durée de consultation, etc.). Celles-ci sont évidemment plus simples à obtenir étant donné l'absence de sollicitation des utilisateurs mais leur fiabilité est moindre [Isinkaye et al., 2015]. De plus, des difficultés comme les biais potentiels, les lacunes (comment inférer des évaluations négatives ?) ou l'interprétation des comportements sont réelles [Ricci et al., 2015].

Dans le cas de la recommandation de littérature scientifique, il existe plusieurs manières d'inférer des transactions implicites à partir des données laissées par un utilisateur potentiel. La plus fréquente est sans doute l'utilisation de l'historique des publications comme une représentation des centres d'intérêt [Amami et al., 2016, Sugiyama and Kan, 2015, Yang et al., 2019]. Par extension, l'historique des références citées est parfois utilisé [Amami et al., 2017, Chen and Lee, 2018]. Enfin, une approche plus récente, qui a notamment émergée avec l'apparition des outils de gestion de références bibliographiques en ligne, est l'utilisation de la bibliothèque virtuelle d'un utilisateur, c'est-à-dire l'ensemble des articles pour lesquels ce dernier a marqué son intérêt [Alfarhood and Cheng, 2019, Liu, Yang, Lee, Xu, Yu and Xia, 2015]. Cette dernière approche est cependant moins fréquente étant donné le faible nombre de jeux de données disponibles actuellement (voir section 3.5). Dans tous les cas évoqués, l'utilisation d'une échelle d'évaluation unaire prédomine également.

Combinaison des transactions explicites et implicites

Enfin, les évaluations explicites et implicites peuvent être combinées selon une logique de vérification ou de complémentarité [Isinkaye et al., 2015]. Par exemple, un scénario peut être de demander une évaluation explicite à un utilisateur lorsque celui-ci marque un intérêt implicite pour un item. D'ailleurs, Ricci et al. [2015] souligne l'intérêt de combiner évaluations implicites et explicites afin d'obtenir de meilleurs résultats. Dans le cas de la recommandation de littérature scientifique, la manière la plus fréquente de combiner les évaluations explicites et implicites est d'utiliser un ou plusieurs articles sélectionnés par l'utilisateur et l'historique des publications de celui-ci afin d'affiner son profil [Cai, Han and Yang, 2018, Ma and Wang, 2019, Mu et al., 2018].

2.1.3 Cold-start

Le problème du *cold-start* est l'impossibilité de générer des recommandations fiables à cause d'un manque de données d'évaluation [Bobadilla et al., 2013]. Il concerne les systèmes de recommandation indépendamment du domaine et s'applique donc également à la recommandation de littérature scientifique [Bai et al., 2019]. Ce problème est d'ailleurs une des principales motivations de l'utilisation de transactions implicites afin d'obtenir un corpus directement prêt à l'emploi [Ricci et al., 2015]. Trois variantes du *cold-start* sont distinguées [Bobadilla et al., 2013] : la nouvelle communauté, le nouvel item et le nouvel utilisateur.

La nouvelle communauté concerne le manque de transactions explicites dû au déploiement récent d'un système de recommandation. Ce dernier est donc dans l'incapacité de s'appuyer sur ces données pour générer des recommandations, quelque soit l'item ou l'utilisateur concerné. Une solution fréquemment employée dans ce cas est l'utilisation de méthodes de recommandation ne nécessitant pas ou peu de données d'évaluation explicite, par exemple des méthodes basées sur le contenu des items (voir section 2.2.3). Il peut s'agit d'une solution provisoire le temps de récolter suffisamment de transactions.

Le nouvel item est un item récemment ajouter au corpus et pour lequel le système ne possède pas encore de transaction enregistrée. Ce qui a pour conséquence que cet item risque de ne jamais être recommandé. De manière générale, cette variante est la moins impactante pour un système de recommandation car elle n'est pas directement préjudiciable pour l'utilisateur [Bobadilla et al., 2013]. Cependant, il est important de veiller à intégrer les nouveaux items afin de garantir le renouvellement des recommandations à plus long terme. Encore une fois, une solution envisageable dans ce cas est d'utiliser des méthodes ne s'appuyant pas sur un corpus de transaction. Dans le cas de la recommandation de littérature scientifique, le nouvel item est malgré tout un problème plus grave car le suivi de la recherche, et donc l'actualité des recommandations, sont des aspects importants pour l'utilisateur.

Enfin, le nouvel utilisateur est un utilisateur pour lequel le système ne possède pas de transaction et ne peut donc pas générer des recommandations fiables. Il s'agit d'une variante beaucoup plus problématique car elle nuit fortement au ressenti de l'utilisateur par rapport à l'utilité du système. La solution la plus évidente est de demander explicitement à l'utilisateur de soumettre une sélection d'items avec évaluations, ce qui est d'ailleurs fréquemment le cas dans les systèmes de recommandation de littérature scientifique. Il existe également d'autres stratégies basées sur l'active learning qui consistent à demander à l'utilisateur d'évaluer certains items proposés par le système afin de pouvoir générer des recommandations [Ricci et al., 2015]. Une autre solution envisageable est d'utiliser des données spécifiques sur l'utilisateur lui-même afin de localiser des utilisateurs similaires pour lesquels le système possède des transactions. Celles-ci peuvent dès lors être utilisées pour générer des recommandations [Khusro et al., 2016].

2.1.4 Au delà des transactions utilisateurs-items

L'utilisateur peut également être modélisé au-delà de ses transactions avec les articles afin d'améliorer la qualité des recommandations. Diverses caractéristiques comme des données démographiques ou associées à la profession peuvent ainsi être rassemblées [Ricci et al., 2015]. Ou encore différents profils, besoins ou buts peuvent être définis [Beel et al., 2016]. Cependant, il s'agit d'un aspect plutôt négligé dans le développement des méthodes de recommandation en littérature scientifique. Alors que la modélisation de l'utilisateur est essentielle pour améliorer l'efficacité des systèmes de recommandation [Beel et al., 2016]. Il est important tout de même d'être attentif au coût cognitif pour l'utilisateur consécutif à l'acquisition de ces informations, notamment en le mettant en balance avec le gain qu'il peut en espérer [Ricci et al., 2015].

Un exemple de méthode s'appuyant sur un type de profil est proposé par West et al. [2016]. Outre le choix explicite d'un article d'intérêt, l'utilisateur se voit proposer trois profils types afin d'orienter les recommandations. Le profil *classique* favorise des articles de référence parmi une large sélection d'articles sémantiquement proches de l'article d'intérêt. Le profil *expert* favorise également les articles de référence mais parmi une sélection plus restreinte. Et enfin, le profil *sérendipité* propose une sélection aléatoire de références parmi un ensemble restreint d'articles sémantiquement proches de l'article d'intérêt.

Un autre exemple est proposé par Dhanda and Verma [2016]. Cette méthode permet à l'utilisateur de paramétrer une fonction d'utilité selon divers critères comme l'autorité des articles ou la date de publication. Elle fonctionne en deux étapes. La première consiste en la sélection d'un ensemble d'articles sémantiquement proche d'un sujet choisi explicitement par l'utilisateur (celui-ci pouvant être éventuellement dérivé d'un article d'intérêt). Et la seconde s'appuie sur cette fonction

d'utilité afin de classer les articles candidats.

Ricci et al. [2015] évoque aussi la distinction entre préférences à long terme (par exemple, l'historique des publications ou la bibliothèque de l'utilisateur) et à court terme (par exemple une sélection explicite et ponctuelle de quelques articles). Cette distinction est importante pour évaluer la pertinence des recommandations pour l'utilisateur. Beel et al. [2016] souligne d'ailleurs que l'utilisation de préférences à long terme, notamment via l'historique des publications de l'utilisateur, n'est pas toujours idéal pour décrire son besoin informationnel. Il suggère plutôt l'utilisation d'une fenêtre temporelle adéquate.

Un exemple plutôt original combinant les deux types de préférences est proposé par Zhao et al. [2016]. D'une part, l'utilisateur définit un sujet d'intérêt à court terme via un petit texte comme un projet de recherche ou un abstract. Ce sujet est considéré par la méthode comme représentatif des connaissances que l'utilisateur souhaite acquérir. Ensuite, l'historique des articles lus constitue les préférences à long terme et est considéré comme étant représentatif des connaissances déjà acquises par l'utilisateur. Et la méthode propose des recommandations permettant de faire le lien entre connaissances acquises et à acquérir, afin de proposer un parcours d'apprentissage à l'utilisateur.

2.2 Prédiction de l'évaluation

2.2.1 Présentation formelle et panorama des méthodes

La deuxième étape du processus de recommandation consiste à prédire les scores d'évaluation des items du corpus pour les différents utilisateurs à partir des préférences collectées précédemment. Formellement, il s'agit donc de construire une fonction $\hat{r} : \mathcal{U}, \mathcal{I} \rightarrow \mathcal{R}$ permettant de générer des transactions $(u, i, \hat{r}(u, i))$ lorsque r n'est pas définie, c'est-à-dire qu'une transaction impliquant l'utilisateur u et l'item i n'a pas pu être collectée [Ricci, 2017]. Il s'agit de la première version du problème de la recommandation qui consiste donc à prédire les évaluations manquantes [Aggarwal, 2016].

Plutôt que de prédire des évaluations, la seconde version envisage le problème de la recommandation comme un problème de classement qui peut être également résolu via le classement des n meilleurs items pour chaque utilisateur [Aggarwal, 2016]. Il s'agit alors de produire des évaluations relatives et non plus absolues comme dans le premier cas. Ces scores d'évaluation n'ont dès lors plus de lien avec une quelconque échelle et ont uniquement pour but d'ordonner les items. Cette manière d'envisager le problème de la recommandation donne naissance à une nouvelle classe de méthodes.

La classification proposée ici (voir table 2.1) a simplement pour but d'organiser la présentation de la phase de prédiction de l'évaluation. Les classes et les approches définies ne doivent donc pas être vues comme des catégories strictes. À noter qu'il existe de nombreuses autres manières de catégoriser les méthodes de recommandation en général [Aggarwal, 2016, Isinkaye et al., 2015] et en littérature scientifique [Bai et al., 2019, Beel et al., 2016].

2.2.2 Filtrage collaboratif

Le principe générale de cette classe est de partir des transactions entre utilisateurs et items pour générer des recommandations. Elle repose sur l'hypothèse que des utilisateurs similaires auront tendance à évaluer les items de manière similaire [Aggarwal, 2016]. Concrètement, les transactions

TABLE 2.1 – Récapitulatif des classes et approches de recommandation

Classe	Description	Approches
Filtrage collaboratif	utilisation des transactions sous forme d'une matrice utilisateurs-items	items ou utilisateurs similaires, construction d'un modèle
Recommandation basée sur le contenu	utilisation du contenu et des méta-données des items	
Recommandation basée sur la connaissance	modélisation des préférences de l'utilisateur sous forme de spécifications	contraintes, exemples
Recommandation basée sur les graphes	représentation du domaine sous forme de graphe	
Approches globales	pas d'utilisation des transactions spécifiques de l'utilisateur cible	pertinence globale, données démographiques, stéréotypes
Filtrage hybrique	combinaison de plusieurs méthodes	combinaison de caractéristiques, méta-niveau, augmentation des caractéristiques, cascade, combinaison pondérée des recommandations, alternance entre méthodes
Approches basées sur le classement	recommandation vue comme un problème de classement	*pointwise*, *pairwise*, *listwise*

prennent la forme d'une matrice d'évaluation R de dimensions $m \times n$, m étant le nombre d'utilisateurs et n le nombre d'items, et où chaque élément $R_{i,j}$ correspond au score d'évaluation de l'item j par l'utilisateur i. Et les méthodes de cette classe prédisent les évaluations manquantes en exploitant cette dernière. Il existe deux ensembles principaux d'approches [Aggarwal, 2016] : les approches basées le voisinage (ou la mémoire) qui exploitent directement les données connues (i.e. la matrice des évaluations), et les approches basées sur les modèles ou l'idée est de construire un modèle prédictif à partir de la matrice d'évaluation.

Approches basées sur le voisinage

La première approche basée sur le voisinage consiste à rechercher des utilisateurs similaires par le biais de leurs évaluations respectives (i.e. les lignes dans la matrice d'évaluation). Les utilisateurs les plus similaires constituent le voisinage et sont alors utilisés pour estimer les évaluations des items inconnus de l'utilisateur cible. Différentes mesures de similarité peuvent être employées mais les plus fréquentes sont le coefficient de corrélation de Pearson et la similarité cosinus [Isinkaye et al., 2015]. Les évaluations des items inconnus de l'utilisateur cible sont ensuite estimées via la moyenne des évaluations pondérées par la similarité des utilisateurs avec l'utilisateur cible. À noter que d'autres techniques pour combiner les évaluations existent [Melville and Sindhwani, 2017]. Une méthode utilisant cette approche est proposée par Liu, Kong, Bai, Wang, Bekele and Xia [2015]. Elle utilise une matrice d'évaluation où les utilisateurs sont les articles citant d'autres articles, les items sont les articles cités, et les scores d'évaluation sont basés sur les citations et suivent une échelle binaire.

La seconde approche basée sur le voisinage est analogue à la première mais avec des items similaires. L'idée est que pour chaque item non évalué par l'utilisateur cible, le système recherche parmi

les items de l'utilisateur cible ceux ayant des évaluations similaires (i.e. les colonnes de la matrice d'évaluation). Les items les plus proches (i.e. le voisinage) sont utilisés pour estimer l'évaluation manquante. Le filtrage basé sur les items est en général plus précis car il se base sur les évaluations des items de l'utilisateur [Aggarwal, 2016] mais le filtrage via utilisateurs similaires aurait tendance à proposer des recommandations plus originales [Ricci et al., 2015]. À noter également que les deux approches peuvent être combinées pour augmenter la précision des prédictions [Aggarwal, 2016]. Un exemple s'inspirant de cette dernière idée appliquée à la littérature scientifique est d'ailleurs proposé par Haruna et al. [2017].

Approches basées sur les modèles

Les approches par voisinage posent cependant des problèmes liés à la matrice d'évaluation [Aggarwal, 2016]. D'une part, sa taille occasionne des difficultés en terme de mémoire et de calcul. Et d'autre part le nombre parfois important de valeurs manquantes peut rendre difficile le calcul de certaines estimations. Bien que des techniques de clustering ou de recherche approchée peuvent être employées, le manque de valeurs reste problématique.

Les méthodes basées sur les modèles offrent souvent des avantages par rapport aux précédentes comme la moindre complexité en espace, la plus grande vitesse d'entraînement et de prédiction, et le moindre risque de sur-spécialisation [Aggarwal, 2016]. La principale approche est l'utilisation de modèles à facteurs latents, c'est-à-dire des techniques de réduction de dimensionnalité appliquées à la matrice d'évaluation [Aggarwal, 2016]. La technique la plus souvent citée est la décomposition en valeurs singulières (et ses variantes *Funk MF* et *SVD++*. Mais d'autres peuvent être utilisées, comme les méthodes de factorisation *Alternate Least Square* employée par Chen and Lee [2018], *Probalistic Matrix Factorization* employée par Alfarhood and Cheng [2019] ou encore *Low-rank and Sparse Matrix Factorization* employée par Dai, Gao, Zhu, Cai and Pan [2018].

Enfin, d'autres modèles peuvent également être utilisés comme des arbres de décision (ou de régression), des règles d'association (qui sont particulièrement intéressantes lorsque les évaluations sont unaires ou binaires), des classifieurs bayésiens naïfs et des réseaux de neurones ou des techniques de complétion de la matrice d'évaluation [Aggarwal, 2016, Isinkaye et al., 2015].

Conclusion

De manière générale, l'intérêt majeur des approches collaboratives est qu'elles sont indépendantes du domaine [Isinkaye et al., 2015]. Une méthode générique peut être employée pratiquement directement à partir du moment où l'on dispose de transactions entre utilisateurs et items, à l'image de Ortega et al. [2018] qui utilise un framework de recommandation générique pratiquement tel quel. Elles sont également intéressantes lorsque les méta-données des items sont relativement pauvres [Isinkaye et al., 2015], bien que ce ne soit pas vraiment le cas dans le domaine de la littérature scientifique. Les inconvénients principaux de cette classe sont le problème du *cold-start*, la sparsité de la matrice d'évaluation et la scalabilité [Isinkaye et al., 2015].

Cette classe est peu explorée dans le cadre de la recommandation de littérature scientifique [Beel et al., 2016]. La cause principale est la difficulté d'obtenir des évaluations explicites des items par les utilisateurs. Afin de pallier ce problème, certaines approches infèrent les évaluations à partir des interactions des utilisateurs avec les items ou des citations (avec les inconvénients associés : biais, mauvaise interprétation, perte de qualité, etc.). Bai et al. [2019] nuance le problème de l'accès à des données d'évaluation de manière générale en soulignant la disponibilité de nouveaux jeux de

données liés aux réseaux sociaux scientifiques comme *CiteULike*. Il souligne d'ailleurs l'émergence de techniques basées sur la similarité des auteurs via des données en provenance des réseaux sociaux.

Cependant, Beel et al. [2016] considère que le problème général de cette classe est la sparsité de la matrice d'évaluation causée par le nombre important d'items pour peu d'utilisateurs (par opposition à la recommandation de films par exemple), ce qui implique que trouver des utilisateurs ou des items similaires n'est pas évident, sans parler des inconvénients en terme scalabilité.

2.2.3 Recommandation basée sur le contenu

La recommandation basée sur le contenu consiste à exploiter le contenu des items pour générer des recommandations. L'idée est d'extraire des caractéristiques à partir des items déjà évalués par l'utilisateur afin de lui proposer des items similaires. Les deux sources de données nécessaires sont donc les méta-données relatives aux items et les données du profil de l'utilisateur à qui sont destinées les recommandations [Aggarwal, 2016]. Cette classe de recommandation est surtout intéressante lorsque les items possèdent de nombreuses caractéristiques exploitables pour les distinguer, ce qui est le cas des items textuels comme les publications scientifiques [Aggarwal, 2016]. À la différence du filtrage collaboratif, la conception des méthodes appartenant à cette classe nécessite une connaissance spécifique du domaine de recommandation, ce qui rend les méthodes plus difficilement transposables dans un autre domaine [Isinkaye et al., 2015]. Enfin, les méthodes de cette classe procède en général en trois phases [Aggarwal, 2016, Bai et al., 2019] : le preprocessing et l'extraction des caractéristiques des items, la construction du profil de l'utilisateur cible, et la prédiction des évaluations.

Preprocessing et choix des méta-données

Le preprocessing consiste à sélectionner et à préparer les méta-données nécessaires au calcul des représentations. La plupart du temps, les méthodes se focalisent sur l'exploitation du titre et de l'abstract [Beel et al., 2016], notamment parce que ces méta-données sont fréquemment disponibles et qu'elles offrent une bonne représentativité du contenu des articles [Chen, 2010]. Néanmoins, d'autres méthodes utilisent parfois en complément les mots-clés disponibles, qu'ils soient choisis librement par les auteurs ou tirés d'un référentiel spécifique à une base de données particulière (par exemple le *Medical Subject Headings* (MeSH) pour la base de données MEDLINE [1]). Les mots-clés ne sont cependant pas toujours présents. Néanmoins, des méthodes d'extraction à partir des titres et des abstracts peuvent pallier leur absence [Chakraborty et al., 2015]. Les textes complets sont parfois employés mais tant les restrictions en terme d'accessibilité que les difficultés en terme d'exploitation de documents beaucoup plus longs et complexes rendent ce choix également plus rare [Khabsa and Giles, 2014]. Enfin, il est possible de prendre en compte le pouvoir discriminant des différents champs (selon l'ordre d'importance : titre $>$ abstract $>$ texte complet) mais cette stratégie est également peu employée [Beel et al., 2016].

Le calcul des représentations consiste à transformer les méta-données afin de faciliter la comparaison entre les articles tout en préservant les informations contenues. Les techniques employées proviennent principalement du traitement automatique des langues ou *natural language processing* (NLP).

Une première approche est l'utilisation d'ensembles de mots. Elle n'est pas vraiment performante avec des données textuelles comme le titre ou l'abstract [Kowsari et al., 2019]. Elle peut néanmoins

1. `https://www.nlm.nih.gov/mesh/meshhome.html`

être employée avec des mots-clés, ce qui est le cas de Wang et al. [2017] par exemple, mais son emploi reste malgré tout très marginal dans la littérature.

L'approche classique est l'utilisation de sacs-de-mots (ou *bags-of-words*). L'idée est de transformer chaque document en un vecteur V de longueur équivalente au nombre de mots distincts composant le vocabulaire du corpus (i.e. l'ensemble des documents), et dont chaque élément V_i est une valeur relative à la fréquence d'apparition de ce mot dans le document. V_i peut être calculé de plusieurs façons mais les méthodes prenant en compte la fréquence des mots dans un document et la fréquence de ces mots dans le corpus sont les plus utilisées. TF-IDF [Jones, 1972] est la technique la plus fréquemment employée [Bai et al., 2019, Beel et al., 2016], par exemple dans Lee et al. [2015], Bulut et al. [2018], Wang et al. [2018] ou encore le système de recommandation *Arxiv-Sanity*[2]. BM25 [Robertson and Zaragoza, 2010] est également souvent employée en recherche d'information de manière générale, et souvent préférée à TF-IDF comme référence pour évaluer d'autres méthodes. Elle est également intéressante pour le passage en production car elle est implémentée dans certains composants populaires comme la librairie *Apache Lucene*[3] ou le serveur de recherche *ElasticSearch*[4]. Un exemple de cette utilisation est proposée par Mohamed Hassan et al. [2019].

Une autre approche est l'utilisation de techniques d'extraction de thèmes latents. L'allocation de Dirichlet latente ou *latent Dirichlet allocation* (LDA) [Blei et al., 2003] est sans doute la technique la plus fréquemment utilisée [Beel et al., 2016]. L'idée est de construire un modèle générateur de documents à partir d'un corpus de documents et un nombre de sujets défini en paramètre. Ce modèle repose sur l'hypothèse qu'un document est composé d'un certain nombre de sujets selon une distribution de Dirichlet, et que chaque sujet est composé de mots selon une certaine probabilité d'apparition. Elle est notamment employée par Amami et al. [2016], qui utilise la mesure de perplexité pour déterminer le nombre optimal de sujets.

Une autre approche est l'utilisation de techniques de plongement lexical ou *word embedding*. Il s'agit de techniques permettant de représenter un ensemble de mots dans un espace vectoriel de telle manière que les positions relatives des différents mots soient sémantiquement significatives. La technique la plus fréquente est Word2Vec [Mikolov et al., 2013]. Ce modèle repose sur l'hypothèse qu'un mot est défini par son contexte, c'est-à-dire les mots à proximité. Il est donc constitué d'un réseau de neurones entraîné à prédire un mot à partir de son contexte (variante *continuous bag-of-words*) ou inversément à prédire un contexte à partir du mot associé (variante *skip-gram*). Afin de représenter un document, il faut cependant combiner les vecteurs obtenus. Plusieurs techniques existent comme la simple moyenne ou la moyenne pondérée par la fréquence d'apparition. D'autres techniques plus élaborées peuvent également être employées comme Doc2Vec [Le and Mikolov, 2014].

Enfin, les récentes avancées en deep learning permettent de disposer de nouvelles techniques. Un premier ensemble est lié à l'utilisation des réseaux de neurones récurrents. Par exemples, les méthodes proposées par Ravi et al. [2017] et Yang et al. [2018] utilisent des réseaux de type LSTM. Un seconde ensemble consiste à utiliser un modèle de langage comme BERT (*Bidirectional Encoder Representations from Transformers*) [Devlin et al., 2018]. Mohamed Hassan et al. [2019] utilise différents modèles (USE, BERT, InferSent, ELMO et SciBERT) mais les premiers résultats obtenus ne semblent guère convaincants et les coûts en calcul pour obtenir les représentations sont assez

2. `http://www.arxiv-sanity.com/`
3. `https://lucene.apache.org/`
4. `https://www.elastic.co/`

importants.

Construction du profil de l'utilisateur

La construction du profil utilisateur est généralement réalisée à partir d'une sélection d'articles d'intérêt en exploitant les représentations obtenues lors de la phase précédente [Bai et al., 2019]. Dans le cas ou l'utilisateur est caractérisé par un seul article, sa représentation est généralement utilisée telle quelle. Par contre, dans le cas de plusieurs articles, différentes approches sont possibles pour mettre en commun les représentations de ceux-ci [Bai et al., 2019]. Il est également possible de maintenir les représentations distinctes et de combiner les prédictions obtenues au moment de la recommandation (comme dans Lee et al. [2015]).

Prédiction des évaluations

Une fois les représentations des documents calculées et le profil de l'utilisateur cible construit, il ne reste qu'à prédire les évaluations. L'approche la plus simple et également la plus fréquente est la similarité cosinus entre le profil de l'utilisateur et les articles du corpus [Bai et al., 2019, Beel et al., 2016], employée par exemple dans Bulut et al. [2018].

Une autre approche est l'utilisation d'un classifieur. Le séparateur à vaste marge ou *Support vector machine* (SVM) est une technique fréquemment employée [Beel et al., 2016]. Par exemple, *Arxiv-Sanity* emploie un SVM entraîné à reconnaître les articles pertinents à partir des articles du profil de l'utilisateur cible (sous forme de vecteurs TF-IDF). Un autre exemple est la méthode proposée par Wang et al. [2018] qui utilise une régression *softmax* comme classifieur. Enfin, Bhagavatula et al. [2018] emploie un réseau de neurones entraîné à estimer des probabilités qu'un article cite un autre à partir de paires d'articles sous forme de représentations *bag-of-words*.

Conclusion

Il s'agit d'une des classes les plus explorées dans le domaine de la recommandation de littérature scientifique [Beel et al., 2016]. Outre le fait d'être particulièrement adaptées aux items riches en contenu exploitable, les méthodes appartenant à cette classe comportent d'autres avantages comme la facilité pour recommander de nouveaux items (i.e. pour lesquels il n'y a pas ou peu de transactions) et l'explicabilité des recommandations [Aggarwal, 2016]. De plus, contrairement au filtrage collaboratif, elles ne nécessitent pas de données relatives aux autres utilisateurs [Aggarwal, 2016]. Enfin, elles peuvent en général adapter rapidement les recommandations en fonction des changements d'intérêts de l'utilisateur au fil du temps comme dans les méthodes proposées par Chen and Ban [2016] et Zequn Gao [2015].

Ces méthodes comportent néanmoins quelques inconvénients. Elles nécessitent de disposer de méta-données riches sur les items [Isinkaye et al., 2015]. Le coût pour calculer les représentations peut parfois être important [Beel et al., 2016]. Elles ne résolvent pas le problème du *cold-start* pour les nouveaux utilisateurs [Aggarwal, 2016]. Les recommandations générées ont souvent une faible capacité à surprendre l'utilisateur et le risque de sur-spécialisation n'est pas négligeable [Beel et al., 2016]. Enfin, les recommandations ne prennent pas en compte des critères comme la qualité, la complexité ou l'autorité des articles recommandés [Bai et al., 2019, Beel et al., 2016].

2.2.4 Recommandation basée sur la connaissance

Cette classe utilise les connaissances spécifiques au domaine pour proposer des recommandations [Ricci et al., 2015]. L'idée est d'identifier les caractéristiques susceptibles de rencontrer les besoins et les préférences des utilisateurs. Cette classe est similaire à la recommandation basée sur le contenu à la différence que l'utilisateur exprime explicitement les spécifications nécessaires pour les recommandations [Aggarwal, 2016]. Les entrées sont donc les spécifications, les attributs des items et la connaissance du domaine. Cette classe est adaptée dans les situations suivantes [Aggarwal, 2016] : l'utilisateur souhaite spécifier explicitement ses besoins, l'obtention des évaluations pour les items est difficile à cause de la complexité du domaine des items (nombreux types et variantes disponibles), et les évaluations sont sensibles au temps.

Le fonctionnement des méthodes de cette classe repose généralement sur quatre ensembles de fonctionnalités [Felfernig et al., 2015] :

— la collecte des besoins de l'utilisateur,
— la recommandation sur base de la connaissance des items et de leur bonne correspondance avec les besoins de l'utilisateur,
— la gestion des éventuelles inconsistances entre les besoins de l'utilisateur et les items disponibles (c'est-à-dire le cas où aucun item ne peut être proposé),
— et les explications sur le processus de recommandation.

Approches

La première approche est la recommandation basée sur les contraintes. L'idée est que l'utilisateur exprime ses besoins sous forme de contraintes afin de guider le processus de recommandation. Cette approche est particulièrement adaptée lorsqu'il est facile d'exprimer explicitement des contraintes sur les caractéristiques des items (par exemple sur une date de publication) [Aggarwal, 2016]. La recommandation est alors exprimée comme un problème de satisfaction de contraintes exploitant une base de connaissances comprenant les items, leurs caractéristiques et leurs liens avec les utilisateurs [Felfernig and Burke, 2008]. Cette base peut être construite notamment via l'extraction automatique de données à partir des items ou encore par l'intermédiaire de la communauté des utilisateurs [Felfernig et al., 2015]. Les items candidats sont ensuite ordonnés par l'intermédiaire d'une fonction d'utilité [Felfernig and Burke, 2008].

La deuxième approche est la recommandation basée sur les exemples. L'utilisateur exprime cette fois ses besoins sous forme d'exemples (i.e. des items) qui servent de points d'ancrage pour orienter les recommandations [Aggarwal, 2016]. Cette approche est adaptée lorsqu'il est difficile d'exprimer explicitement des contraintes (par exemple, un sujet d'intérêt) [Aggarwal, 2016]. Elle repose sur la combinaison de deux fonctions [Aggarwal, 2016] : une fonction de similarité ou d'utilité permettant de récupérer des recommandations, et une fonction de critique permettant d'explorer interactivement l'espace des items selon les différents buts de l'utilisateur. Cette notion de critique est assez féconde et est d'ailleurs exploitée par des méthodes de recommandation spécifiques [Chen and Pu, 2012]. La stratégie de recommandation employée procède en plusieurs étapes [Bridge et al., 2005] : soit en générant de nouvelles recommandations à chaque fois, soit en réduisant l'ensemble des items candidats.

Ces approches sont également caractérisées par des modes d'interaction propres et pouvant prendre différentes formes [Aggarwal, 2016, Bridge et al., 2005, Felfernig et al., 2015] :

— un système conversationnel basé sur une boucle de feedback interactive pour construire les

spécifications,
- la recherche via un jeu de questions ou un formulaire permettant de déterminer les spécifications,
- et la navigation par manipulation d'une liste de résultats afin d'affiner les spécifications.

L'utilisation d'une stratégie itérative d'interaction est également fréquente.

Une dernière approche plus marginale est l'utilisation d'une ontologie [Lu et al., 2015]. L'idée est de construire une ontologie représentant formellement la connaissance d'un domaine à partir de concepts et de relations. Celle-ci peut ensuite être utilisée pour évaluer la proximité sémantique des items avec les besoins de l'utilisateur. Par exemple, Neethukrishnan and Swaraj [2017] propose une méthode se basant sur une ontologie générale construite à partir de la taxonomie ACM [5] pour proposer des articles sémantiquement proches des besoins de l'utilisateur.

Conclusion

Cette classe n'est pratiquement pas explorée dans le domaine de la recommandation en littérature scientifique. La raison principale est sans doute qu'elle correspond à des scénarios d'utilisation assez spécifiques. En effet, la motivation derrière le développement d'un système de recommandation dans le domaine de la littérature scientifique est de proposer un service complémentaire à une application principale comme un moteur de recherche spécialisé ou un gestionnaire de références bibliographiques. Ce système exploite les données produites par les utilisateurs lors de leurs interactions avec le système principale pour générer des recommandations. Le rôle de l'utilisateur est donc tout à fait passif vis-à-vis de ce processus de recommandation. A contrario, cette classe est plutôt destinée à des scénarios de recommandation ou l'utilisateur est actif. Le schéma d'interaction, souvent itératif et demandant à l'utilisateur de spécifier explicitement ses besoins, est davantage proche du schéma d'interaction avec un moteur de recherche que des scénarios classiques de recommandation.

Même si cette classe demande un plus grand investissement de la part de l'utilisateur, il ne faut pas l'écarter pour autant. En effet, la possiblité de spécifier finement les besoins et la capacité de contrôler le processus de recommandation sont des atouts indéniables. De plus, un autre intérêt de la plupart des méthodes de cette classe est la capacité à pouvoir expliquer les recommandations [Aggarwal, 2016].

2.2.5 Recommandation basée sur les graphes

Le principe de cette classe est de modéliser le domaine de recommandation sous forme de graphe, ce qui donne accès à un ensemble de techniques associées. Le problème de la recommandation peut notamment intégrer différentes notions comme l'autorité, le contexte (i.e. le voisinage) ou encore l'influence [Aggarwal, 2016]. Et la recommandation peut également être vue comme une tâche de prédiction de liens entre nœuds. Comme dans le cas de la recommandation basée sur le contenu, les méthodes développées, bien qu'utilisant des techniques génériques, sont la plupart du temps spécifiques au domaine. Cette classe est assez bien représentée dans le domaine de la recommandation de littérature scientifique [Beel et al., 2016]. Et elle semble prendre de l'importance dans la production récente.

5. https://www.acm.org/publications/class-2012

Différents types de graphes

Différents types de graphes peuvent être construits à partir des données disponibles, éventuellement collectées depuis différentes sources [Bai et al., 2019]. Le plus évident est le graphe bipartite ayant pour nœuds les utilisateurs et les items, et dont les liens sont les transactions entre utilisateurs et items (éventuellement pondérés par les scores d'évaluation) [Lu et al., 2015]. Les relations peuvent être dérivées de différentes sources (par exemple la consultation, la sauvegarde dans sa bibliothèque ou le fait d'être auteur) à l'image des évaluations implicites qui sont dérivées d'interactions diverses [Lu et al., 2015]. Le graphe obtenu est parfois utilisé pour en dériver un autre graphe comme c'est le cas dans les méthodes proposées par Xia et al. [2016] et Liu, Yang, Lee, Xu, Yu and Xia [2015].

Cependant, le graphe le plus fréquemment utilisé dans la littérature est le graphe des citations [Aggarwal, 2016], où chaque article est représenté par un nœud et chaque arête représente une relation de citation. Celui-ci est généralement orienté, bien qu'une variante non-orientée puisse être parfois employée. Enfin, d'autres graphes faisant intervenir les auteurs, les mots-clés, voire les organes de publication peuvent être utilisés. Il existe plusieurs types de graphes hétérogènes en fonction des types de nœuds et des types de relations considérés [Bai et al., 2019, Beel et al., 2016]. De plus, différentes techniques de pondération peuvent être employées. Les graphes hétérogènes sont évoqués dans le filtrage hybride (voir section 2.2.7).

Techniques de partitionnement

Un autre ensemble de techniques fréquemment employées est la recherche de communautés ou le partitionnement (ou clustering). Dans le cas où des relations entre personnes, comme des collaborations, forment le graphe exploité, la motivation est le constat d'une tendance générale à accorder plus de crédit à des recommandations en provenance de connaissances (par exemple, des amis ou des collègues) que des recommandations similaires mais dont l'origine est inconnue [Ricci, 2017]. Il s'agit donc d'exploiter cette notion de confiance entre utilisateurs déduite des relations sociales afin d'améliorer la qualité des recommandations [Lu et al., 2015]. Récemment, l'émergence des réseaux sociaux favorise ce genre d'approches.

Ces techniques ne sont cependant pas limitées aux réseaux sociaux. Elles peuvent être également employées avec d'autres types de graphes. Par exemple, Zhou et al. [2014] applique l'algorithme *Greedy Clique Expansion* sur le graphe des citations afin de diminuer le coût en calcul des recommandations en se limitant au cluster d'intérêt. Une autre méthode proposée par West et al. [2016] utilise la technique de partitionnement hiérarchique *MapEquation* également sur le graphe des citations. La hiérarchie de clusters obtenue permet de faire varier l'ensemble des candidats à la recommandation en fonction des besoins de l'utilisateur cible.

Techniques de classement

Une fois le graphe construit, différentes techniques de classement propres aux graphes sont utilisées pour générer les recommandations. Une approche fréquente est l'utilisation de marches aléatoires [Beel et al., 2016]. La technique la plus fréquemment citée est PageRank. Elle a l'avantage de pouvoir être utilisée dans différents types de graphes [Aggarwal, 2016]. Néanmoins, elle n'est pas toujours la plus adaptée. D'autres variantes peuvent parfois être employées comme *PaperRank* [Bai et al., 2019] ou ALEF (*Article-level Eigenfactor*) [West et al., 2016] qui sont plus adaptées au graphe des citations qui est acyclique.

L'intérêt d'utiliser des marches aléatoires est de pouvoir mesurer l'importance des différents nœuds d'un graphe par l'intermédiaire de leurs liens et de pouvoir notamment intégrer la notion d'autorité d'un nœud, ce qui est particulièrement intéressant dans le cadre de la littérature scientifique. Ces techniques ne peuvent cependant pas proposer de recommandations personnalisées en l'état. Afin, de pallier ce problème, il existe des techniques prenant en compte un ou plusieurs nœuds d'intérêt. L'idée est que le marcheur aléatoire a une certaine probabilité de redémarrer sur ces nœuds d'intérêt à chaque itération. Ce qui a pour conséquence que les parcours des marcheurs aléatoires sont davantage localisés. Deux exemples sont les méthodes proposées par Liu, Yang, Lee, Xu, Yu and Xia [2015] et Xia et al. [2016]. La techniques la plus connue est le PageRank personnalisé [Haveliwala, 2003].

Similarité de voisinage

Une première grande approche exploitant la similarité de voisinage est basée sur la co-occurrence d'items [Beel et al., 2016]. L'idée est de recommander des items qui apparaissent souvent ensemble. Dans le cadre de la littérature scientifique, où cette approche est bien représentée [Beel et al., 2016], cela se fait principalement par le biais des co-citations, bien que d'autres types d'interactions comme les co-téléchargements ou la co-consultation peuvent également être envisagés. L'intérêt de cette approche réside dans le fait qu'elle met en avant les relations entre items plutôt que la similarité de contenu, ce qui favorise la variété des recommandations [Beel et al., 2016]. Un item doit cependant être co-cité au moins une fois pour être un candidat potentiel, ce qui n'est pas toujours le cas et limite donc le nombre d'items potentiellement recommandables. Cette approche renvoie également à certaines mesures de similarité utilisées en scientométrie comme la co-citation ou le couplage bibliographique. Enfin, la notion de similarité de voisinage peut également être étendue à plusieurs niveaux [Son and Kim, 2018].

L'utilisation de critères de similarité de voisinage permet aussi de limiter la portion du graphe prise en compte et donc les calculs nécessaires à la recommandation. Par exemple, la méthode proposée par Hamilton et al. [2017] se base sur les voisins immédiats des articles d'intérêt dans le graphe non-orienté des citations et ne doit donc prendre en compte que les nœuds à une distance de 2 maximum. Son and Kim [2018] utilise également ce critère de localité pour restreindre le nombre de candidats et réduire les calculs nécessaires.

Embeddings

Une approche plus récente est l'utilisation d'*embeddings*. L'idée est de construire des représentations des nœuds d'un graphe dans un espace vectoriel tout en conservant les caractéristiques topologiques de ceux-ci. Les techniques utilisées sont nombreuses et peuvent être regroupées en trois catégories [Goyal and Ferrara, 2018] : la factorisation matricielle, les marches aléatoires et le deep learning. Les techniques les plus fréquemment employées en recommandation de littérature scientifique sont basées sur les marches aléatoires, comme DeepWalk [Perozzi et al., 2014] et Node2Vec [Grover and Leskovec, 2016]. Les représentations obtenues peuvent être comparées via des mesures de similarité comme la similarité cosinus afin d'obtenir des scores d'évaluation. Deux exemples de cette approches sont Tian and Zhuo [2017] et Jia and Saule [2018].

Conclusion

En conclusion, les approches de recommandation basées sur les graphes suscitent beaucoup d'intérêt dans le domaine de la littérature scientifique. Ceci est dû notamment à la possibilité de combiner plusieurs sources d'information différentes et à la richesse des représentations possibles Bai et al. [2019]. La multiplicité des techniques potentiellement employables est également à mentionner, d'autant plus qu'il s'agit d'un domaine de recherche en expansion actuellement. Enfin, l'utilisation de graphes permet également l'introduction de notions comme l'autorité ou la hiérarchie des connaissances qui permettent de répondre plus finement à certains besoins.

Cependant, la question de la scalabilité reste ouverte. En effet, il existe différentes techniques permettant de réduire les calculs comme le clustering. Par contre, d'autres techniques, basées par exemple sur la recherche de plus courts chemins [Son and Kim, 2018], passent plus difficilement à l'échelle. Il est donc important de garder à l'esprit le contexte de production lors de la conception d'une méthode de recommandation appartenant à cette classe.

2.2.6 Approches globales

Les approches globales regroupent les approches qui n'utilisent pas les transactions entre utilisateurs et items pour générer des recommandations. En général, elles n'ont pas beaucoup d'intérêt en tant que telles étant donné l'absence de personnalisation des recommandations produites. Elles peuvent cependant être utiles dans certains scénarios d'utilisation particuliers, notamment lorsque il n'y a pas (encore) de transaction disponible. Elles sont également fréquemment employées en combinaison avec d'autres approches/méthodes.

L'approche basée sur la pertinence globale utilise un ou plusieurs critères communs à tous les utilisateurs, comme la popularité ou la date de publication, pour générer des recommandations [Beel et al., 2016], qui sont donc identiques pour tous les utilisateurs. L'intérêt principale de cette approche est qu'il n'est pas nécessaire de disposer de données relatives aux utilisateurs pour proposer des recommandations. Ce qui peut avoir son intérêt dans certains scénarios de recommandation particuliers, en page d'accueil d'un site web par exemple.

La recommandation basée sur les données démographiques part du constat que différentes recommandations doivent être générées pour différentes niches démographiques [Ricci et al., 2015]. Elle utilise donc les caractéristiques des utilisateurs, comme la langue ou le pays, afin de proposer des recommandations. Cette approche n'est pas vraiment étudiée dans le cadre la littérature sur les systèmes de recommandation [Ricci et al., 2015]. De plus, elle ne semble pas vraiment adaptée au domaine de la littérature scientifique étant donné l'importance et la diversité de celui-ci.

La recommandation basée sur les stéréotypes est une approche inspirée de la psychologie qui permet de catégoriser rapidement des personnes sur base de quelques caractéristiques [Beel et al., 2016]. Cette classe est peu explorée dans le cadre de la recommandation de littérature scientifique et les quelques méthodes existantes ne semblent guère prometteuses [Beel, Dinesh, Mayr, Carevic and Raghvendra, 2017].

2.2.7 Filtrage hybride

Le filtrage hybride regroupe l'ensemble des méthodes qui sont conçues par combinaison de plusieurs classes, approches ou méthodes différentes. La motivation principale derrière cette classe est de profiter des avantages et de limiter les faiblesses des différentes méthodes combinées afin

d'obtenir une méthode plus précise et plus efficace [de Gemmis et al., 2017]. Par exemple, associer une méthode collaborative et une méthode basée sur le contenu est un bon moyen d'atténuer la tendance à la sur-spécialisation de la recommandation basée sur le contenu, et la difficulté pour le filtrage collaboratif de gérer les items et les utilisateurs n'ayant pas ou peu d'évaluations. Il s'agit d'une classe assez bien explorée pour la recommandation de littérature scientifique [Beel et al., 2016], même si ce n'est pas toujours de manière explicite. En effet, beaucoup de méthodes possèdent des caractéristiques hybrides sans pour autant être considérées comme appartenant à cette classe. Une exploration de la littérature montre également que le filtrage hybride suscite de plus en plus d'intérêt dans la recherche récente (voir figure 2.5).

Il existe plusieurs manières de combiner des méthodes de recommandation [Burke, 2002, de Gemmis et al., 2017, Isinkaye et al., 2015]. La taxonomie proposée par Aggarwal [2016] (voir figure 2.2) en identifie trois :

— Les **ensembles** : l'idée est d'utiliser plusieurs méthodes afin d'en obtenir une plus robuste. Les méthodes sont utilisées de manière *black-box*, c'est-à-dire sans modifier leur fonctionnement interne. La première approche consiste à les utiliser parallèlement. Les résultats peuvent alors être combinés via combinaison pondérée des scores ou en alternant la méthode utilisée (*switching*). La seconde approche consiste à utiliser les méthodes de manière séquentielle. Elles peuvent être combinées en cascade si les candidats recommandés par la première méthode sont utilisés en entrées de la suivante. Elles peuvent également être combinées par augmentation de caractéristiques lorsque les prédictions d'évaluation d'une première méthode sont utilisées comme caractéristiques des items par la méthode suivante. Cette dernière approche peut parfois être considérée comme monolithique étant donné qu'il faut modifier le fonctionnement interne des méthodes combinées.
— Le **mélange des recommandations** : les recommandations obtenues via différentes méthodes sont simplement proposées côte-à-côte en même temps à l'utilisateur cible. Cette approche correspond à des cas d'utilisation spécifiques qui sortent du cadre de ce travail.
— La **conception monolithique** : différentes méthodes sont intégrées par modification interne de leur fonctionnement, par opposition aux ensembles et au mélange. La première approche est la combinaison de caractéristiques habituellement utilisées par des méthodes de classes différentes. La seconde est la combinaison à un méta-niveau, c'est-à-dire lorsque l'intrication touche le fonctionnement même des méthodes combinées.

Alternance

L'approche par combinaison de méthodes en parallèle par alternance consiste donc à alterner entre différentes méthodes de recommandation pour prédire les évaluations. La gestion de l'alternance passe par la détermination de critères de sélection souvent liés au contexte de production [de Gemmis et al., 2017]. La motivation principale est d'éviter les difficultés de certaines méthodes dans des situations particulières comme un nouvel utilisateur pour lequel le système ne dispose pas de transaction. L'alternance est d'ailleurs fréquemment employée pour gérer le problème du *cold-start* [Aggarwal, 2016]. Ces critères sont donc liés à l'établissement d'une stratégie prenant en compte les forces et les faiblesses des différentes méthodes [Isinkaye et al., 2015]. Cette approche n'est pas présente dans la littérature sur les méthodes de recommandation en littérature scientifique explorée dans ce travail. Une explication est peut-être que le développement de telles méthodes est surtout lié au contexte de production.

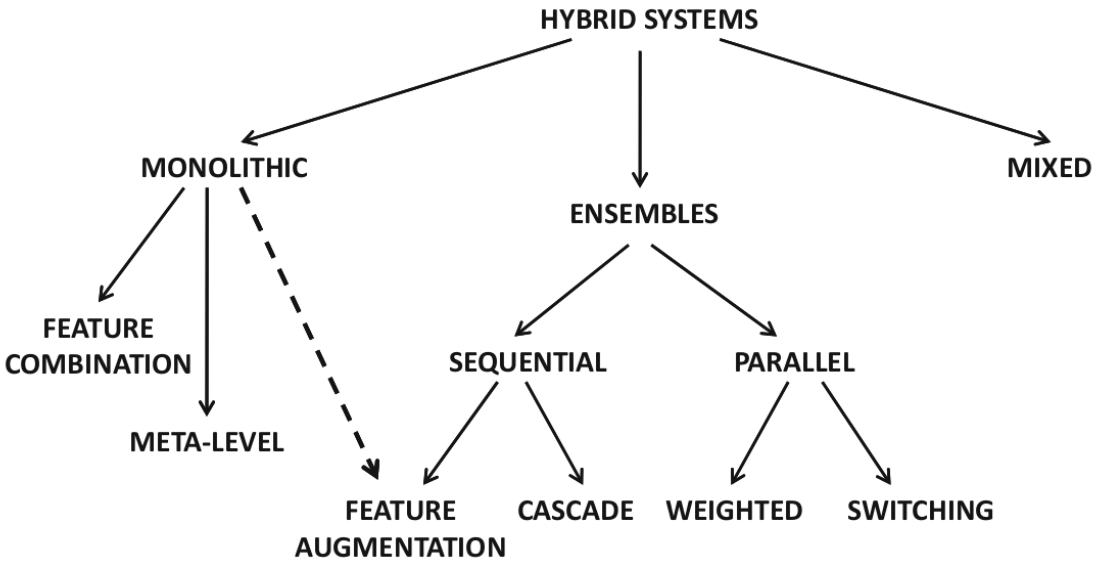

FIGURE 2.2 – Taxonomie des approches hybrides proposée par Aggarwal [2016]

Combinaison pondérée des résultats

La combinaison pondérée des résultats consiste à combiner les scores obtenus par différentes méthodes de recommandation. La technique la plus simple est la combinaison linéaire [de Gemmis et al., 2017] mais d'autres techniques existent. Par exemple, Färber and Sampath [2020] utilise une technique de combinaison stochastique inspirée des algorithmes génétiques. D'autres stratégies de pondération dynamique existent, prenant par exemple en compte les retours de l'utilisateur [Isinkaye et al., 2015]. Qu'il soit statique ou dynamique, le choix des poids passe par une phase d'évaluation ayant pour but de minimiser l'erreur de prédiction [Aggarwal, 2016].

Outre l'amélioration de la précision, cette approche peut être également utilisée lorsqu'une méthode nécessite des données qui ne sont pas toujours disponibles pour l'intégralité des items du corpus utilisé, afin de ne pas limiter les items potentiellement recommandables. Par exemple, Kanakia et al. [2019] part du constat que les citations ne sont pas toujours disponibles et propose une méthode combinant co-citation et contenu afin d'améliorer le nombre de candidats susceptibles d'être pris en compte.

Cascade

L'approche en cascade consiste à combiner séquentiellement plusieurs méthodes de recommandation de telle manière que les items recommandés par une première méthode deviennent les candidats de la méthode suivante. L'idée générale est de construire la liste des recommandations par raffinements successifs [de Gemmis et al., 2017]. Le pattern le plus fréquemment rencontré est de combiner une première méthode dont le but est de dégrossir la liste des candidats potentiels, et une seconde qui raffine le classement selon les besoins de l'utilisateur [Isinkaye et al., 2015]. Un premier exemple de ce pattern est la méthode proposée par Sesagiri Raamkumar et al. [2017]. Celle-ci combine une méthode de recommandation basée sur le contenu permettant d'avoir une sélection de candidats sémantiquement proches du profil de l'utilisateur, avec une méthode basée sur les graphes favorisant la diversité afin d'avoir une liste finale couvrant le sujet d'intérêt de manière large. Un second exemple, proposé par Kong et al. [2018], consiste à utiliser deux modèles de représentation vecto-

rielle (Word2Vec et Struc2Vec) afin d'établir une sélection de candidats. Les candidats obtenus sont ensuite représentés sous forme d'un graphe afin d'entraîner un troisième modèle de représentation vectorielle permettant de prédire les évaluations via calcul de similarité.

Outre l'amélioration du classement, une seconde motivation de cette approche est la scalabilité. L'idée est alors de combiner une première méthode capable de gérer efficacement un corpus d'items de grande taille, avec une seconde méthode offrant des recommandations plus précises mais incapable de gérer un corpus important. Un premier exemple est proposé par Dhanda and Verma [2016] qui utilise une méthode basée sur les graphes proposant un ensemble de candidats via un algorithme de clustering. Cet ensemble est ensuite ordonné grâce à une seconde méthode basée sur le contenu et favorisant la date de publication et le nombre de citations. Un deuxième exemple est proposé par Alshareef et al. [2019] qui limite les candidats via la proximité avec le ou les articles d'intérêt dans le graphe des citations. Un troisième exemple est proposé par Nogueira et al. [2020] Il combine une première méthode basée sur le contenu (BM25) afin de générer une première liste de candidats. Cette liste est ensuite enrichie via une méthode basée sur les graphes afin d'améliorer la couverture. Et une troisième méthode basée sur le contenu (utilisant le modèle de langage BERT [Devlin et al., 2018]) prédit les évaluations finales.

Augmentation des caractéristique

L'approche par augmentation des caractéristiques consiste à utiliser les scores d'évaluation générés par une première méthode de recommandation comme caractéristiques supplémentaires en entrée d'une seconde méthode de recommandation [de Gemmis et al., 2017]. L'objectif est bien sûr d'améliorer la qualité des recommandations finales. L'intérêt de cette approche est de pouvoir augmenter les performances sans modifier fondamentalement le fonctionnement des méthodes de recommandation [de Gemmis et al., 2017]. Par exemple, Sugiyama and Kan [2015] utilise une méthode basée sur le contenu afin d'enrichir le graphe des citations en ajoutant des arêtes lorsque des articles sont similaires. Ce qui permet d'améliorer les performances de la méthode de filtrage collaboratif appliquée sur la matrice d'adjacence de ce graphe. Un autre exemple est l'utilisation par une méthode basée sur le contenu des scores d'une méthode globale afin d'améliorer le classement des items proposés [Beel et al., 2016].

Combinaison de caractéristiques

L'approche par combinaison de caractéristiques consiste à employer des caractéristiques généralement utilisées par des méthodes de recommandation appartenant à des classes différentes. L'idée générale est de combiner des données en provenance de diverses sources en une représentation unifiée avec laquelle employer une méthode de recommandation [Aggarwal, 2016]. Le premier objectif est évidemment d'améliorer les recommandations. Mais dans ce cas-ci, c'est également d'un solution employée pour pallier un manque de données d'un certain type [Isinkaye et al., 2015]. Cette approche est assez bien représentée dans la littérature sur les méthodes de recommandation en littérature scientifique. Ce qui peut être expliqué par la plus grande disponibilité des données, mais également l'émergence de techniques permettant d'intégrer des données hétérogènes comme les graphes ou le deep learning. Les classes les plus fréquemment combinées sont d'ailleurs les recommandations basées sur les graphes et sur le contenu (voir figure 2.4).

Un premier groupe de méthodes utilise des méta-données des items pour créer un graphe des connaissances ou enrichir le graphe des citations en lui ajoutant une dimension « sujets traités ». Les

sujets sont le plus souvent extraits automatiquement du corpus des items. Une première technique fréquemment employée pour l'extraction de sujets est l'allocation de Dirichlet latente (LDA), utilisée notamment par Jardine and Teufel [2014] et Dai, Zhu, Cai, Pan and Yuan [2018] pour enrichir le graphe des citations. Une seconde technique est l'extraction de mots-clés, employée par exemple par De Nart and Tasso [2014] (avec l'algorithme d'extraction *Dikpe KP*) afin de représenter les différents items sous forme de graphes de mots-clés qu'il peut comparer. Enfin, une autre motivation pour construire un graphe des concepts est la possibilité d'étendre la requête initiale, sous forme d'un ou plusieurs articles d'intérêt, afin de prendre en compte les sujets connexes. L'idée étant que la requête initiale peut être biaisée par rapport au sujet d'intérêt réel de l'utilisateur. Liu et al. [2014] applique cette idée via une méthode d'expansion à partir des sujets de départ, et Chakraborty et al. [2015] l'applique via une méthode de clustering.

Un deuxième groupe de méthodes utilise les méta-données disponibles pour construire un graphe hétérogène, c'est à dire un graphe contenant différents types de nœuds et/ou d'arêtes. La variété des graphes possibles est évidemment assez importante. Par exemple, Pan et al. [2015] construit un graphe dont les nœuds sont des mots-clés ou des articles, et les relations sont de types *citation* (entre articles), *similarité sémantique* (entre mots-clés) et *indexation* (entre articles et mots-clés) sur lequel il applique une fonction de similarité pour prédire les scores d'évaluation. Un autre exemple est la méthode proposée par Yang et al. [2019]. Le graphe construit prend en compte les articles, les auteurs et les organes de publication comme nœuds, les arêtes représentant les relations correspondantes. Il entraîne ensuite un modèle de prédiction d'arêtes afin de calculer les scores d'évaluation. Enfin, certaines méthodes s'appuient sur un graphe hétérogène pour entraîner un modèle permettant d'obtenir des représentations vectorielles des articles. La prédiction des scores d'évaluation se fait alors par calcul de similarité entre articles. Un exemple est proposé par Cai, Han, Pan and Yang [2018].

Un troisième groupe de méthodes se caractérise par la construction d'un modèle utilisant le graphe des citations et des données de contenu pour produire des représentations vectorielles des articles, par opposition aux deux premiers groupes où le contenu des articles était utilisé pour construire le graphe. Les données intégrées proviennent généralement du titre et de l'abstract. La prédiction des scores d'évaluation est alors vue également comme un calcul de similarité. Par exemple, Gupta and Varma [2017] utilise une combinaison linéaire des représentations vectorielles des articles via Doc2Vec (titres et abstracts) et DeepWalk (graphe des citations). Les scores d'évaluation sont ainsi calculés par similarité entre les représentations ainsi obtenues. Enfin, il est également possible de combiner modélisation des données sous forme d'un graphe hétérogène et données de contenu pour construire des représentations vectorielles. Cai, Han and Yang [2018], Cai et al. [2019], Zhang et al. [2018] et Brochier [2019] en sont des exemples.

Enfin, le filtrage collaboratif peut également bénéficier de caractéristiques liées au contenu. La manière de faire la plus fréquente consiste à injecter des données de contenu dans le processus de factorisation de la matrice utilisateurs-items. Par exemple, Alfarhood and Cheng [2019] utilise les titres et les abstracts, et Dai, Gao, Zhu, Cai and Pan [2018] utilise les relations entre auteurs et articles. Une méthode plus originale est proposée par Ortega et al. [2018] qui extrait à partir des titres et des abstracts des articles des tuples $< article, sujet, cardinalite >$ où la $cardinalite$ caractérise l'important du sujet dans l'article. Les scores d'évaluation sont calculés à partir de la factorisation de la matrice articles-sujets construite à partir de ces tuples.

Méta-niveau

L'approche par méta-niveau consiste à utiliser le modèle de recommandation lui-même comme entrée pour une autre méthode [Isinkaye et al., 2015]. Par exemple, une première méthode basée sur le contenu peut être employée pour sélectionner des caractéristiques les plus discriminatives pour différents utilisateurs, et une seconde méthode collaborative utilise ces caractéristiques pour trouver des utilisateurs similaires [Aggarwal, 2016].

Dans les méthodes en littérature scientifique, deux exemples similaires sont Ganguly and Pudi [2017] et Ma and Wang [2019]. Il s'agit dans les deux cas d'entraîner un premier modèle à partir des données de contenu et d'utiliser ensuite les représentations vectorielles obtenues comme valeurs d'initialisation d'un second modèle utilisant le graphe des citations. Les prédictions d'évaluation sont obtenues par calcul de similarité entre les représentations finales.

Conclusion

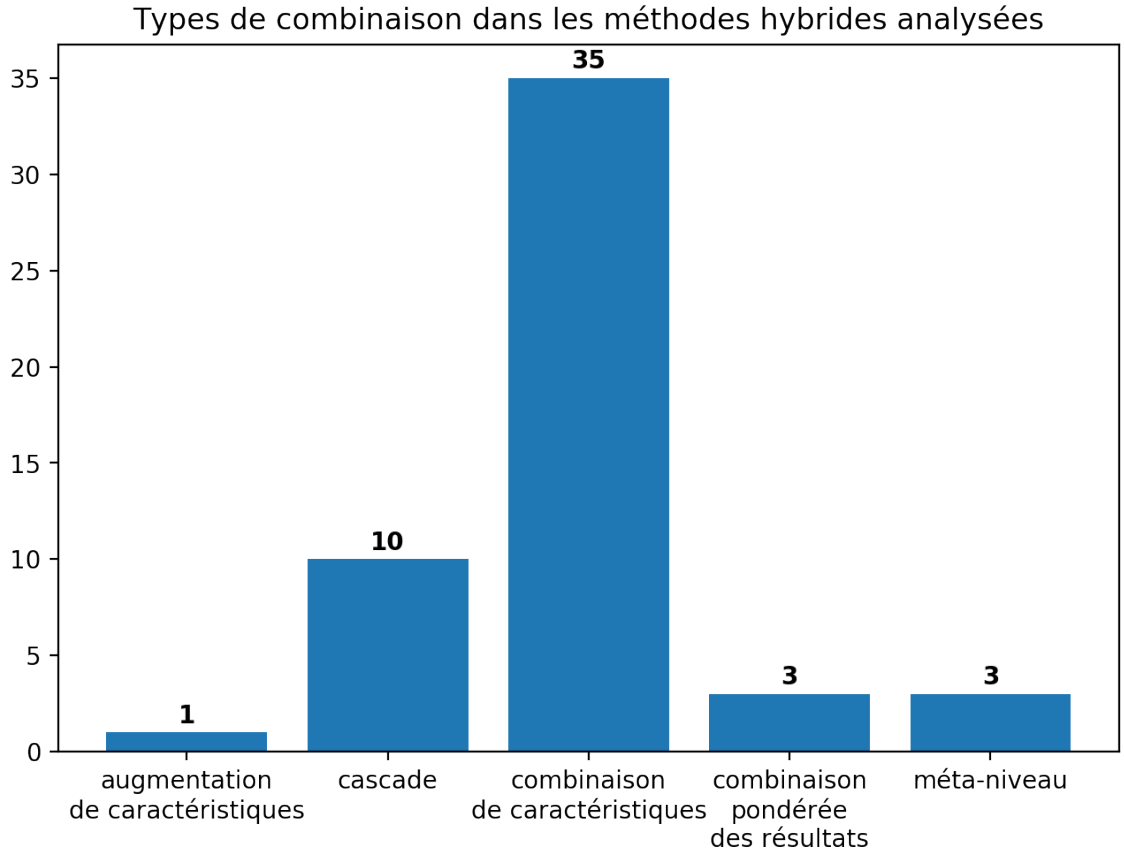

FIGURE 2.3 – Types de combinaison employés dans les méthodes hybrides (voir annexe A.3).

Le filtrage hybride est la classe la plus complexe à mettre en œuvre pour trois raisons : la nécessité de rassembler davantage de données [Isinkaye et al., 2015], la plus forte complexité intrinsèque et la difficulté de trouver une combinaison efficace des méthodes [Bai et al., 2019]. Néanmoins, il s'agit sans doute de la classe de recommandation la plus explorée dans la littérature récente (voir figure 2.5). La motivation principale est sans doute la possibilité d'exploiter des données de différents types afin d'améliorer les recommandations. L'approche la plus fréquente dans ce cas est la combinaison

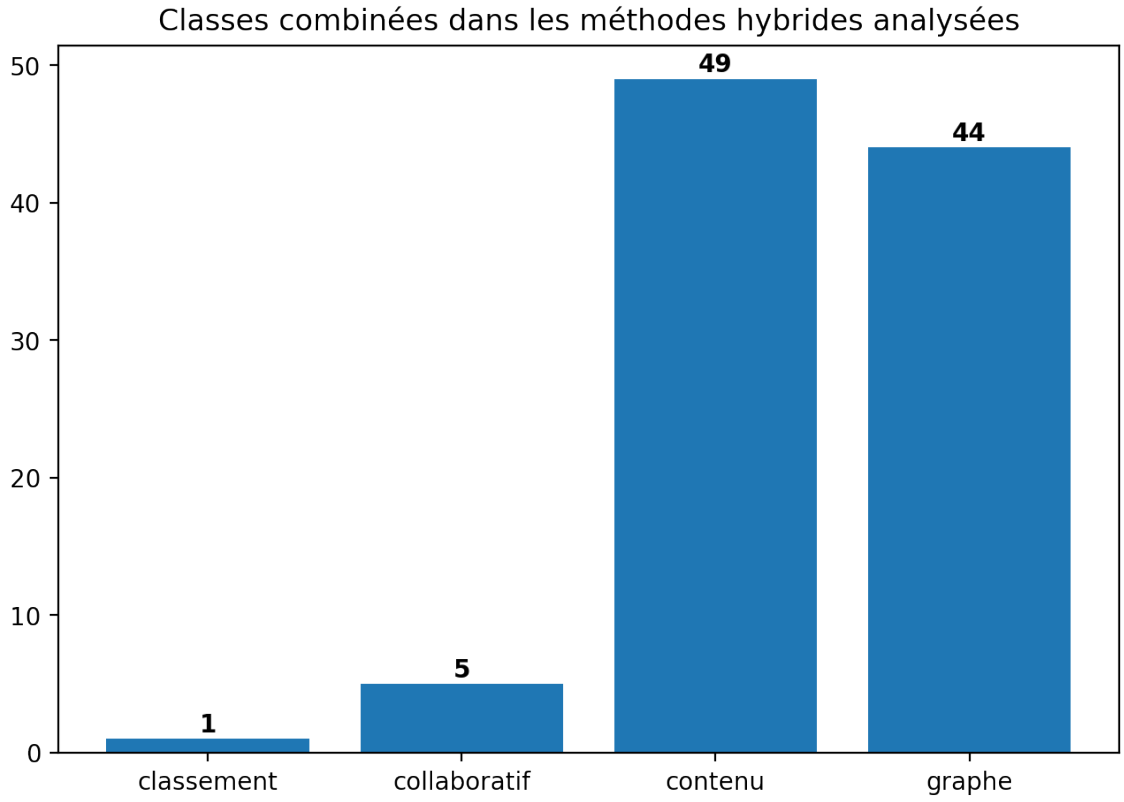

FIGURE 2.4 – Classes combinées dans les méthodes hybrides (voir annexe A.3).

de caractéristiques (voir figure 2.3) en provenance du contenu et des graphes. De manière générale, les méthodes basées sur le contenu et les graphes sont d'ailleurs les plus fréquemment combinées (voir figure 2.4). D'autres enjeux davantage liés à un contexte de production motivent également le développement de méthode hybrides. Par exemple, l'approche en cascade est souvent utilisée pour améliorer la scalabilité.

2.2.8 Approches basées sur le classement

Comme cela a été dit dans la présentation des méthodes (voir section 2.2.1), le problème de la recommandation peut également être envisagé comme un problème de classement. L'idée est alors de proposer à l'utilisateur le classement le plus susceptible de répondre à ses besoins. Les évaluations prédites doivent donc simplement permettre d'ordonner les items. Même si elles découlent de la version prédictive du problème de la recommandation, les classes de recommandation présentées précédemment tendent d'ailleurs à se rapprocher davantage de cette seconde version dans la pratique. Cela se constate notamment dans la manière d'évaluer les méthodes qui considère davantage la précision du classement que celle des évaluations (voir section 3.4).

Les méthodes de cette classe peuvent être regroupées en trois approches selon le type de la fonction objectif employée [Aggarwal, 2016]. L'approche *pointwise* considère les erreurs de prédiction des évaluations pour chaque item. Elle correspond donc à la version prédictive du problème de la recommandation. L'approche *pairwise* utilise des paires d'items pour lesquelles un utilisateur a indiqué s'il préférait le premier ou le second. Et enfin l'approche *listwise* évalue la qualité du classement par le biais de fonctions objectifs comme le gain cumulatif réduit normalisé (*normalized*

discounted cumulative gain) et le rang réciproque moyen (*mean reciprocal rank*).

Les approches basées sur le classement ne sont pas fréquemment employées en recommandation de littérature scientifique. Un exemple est tout de même proposé par Ayala-Gómez et al. [2018] qui utilise une approche *listwise* avec l'algorithme *LambdaMART*.

2.2.9 Conclusion

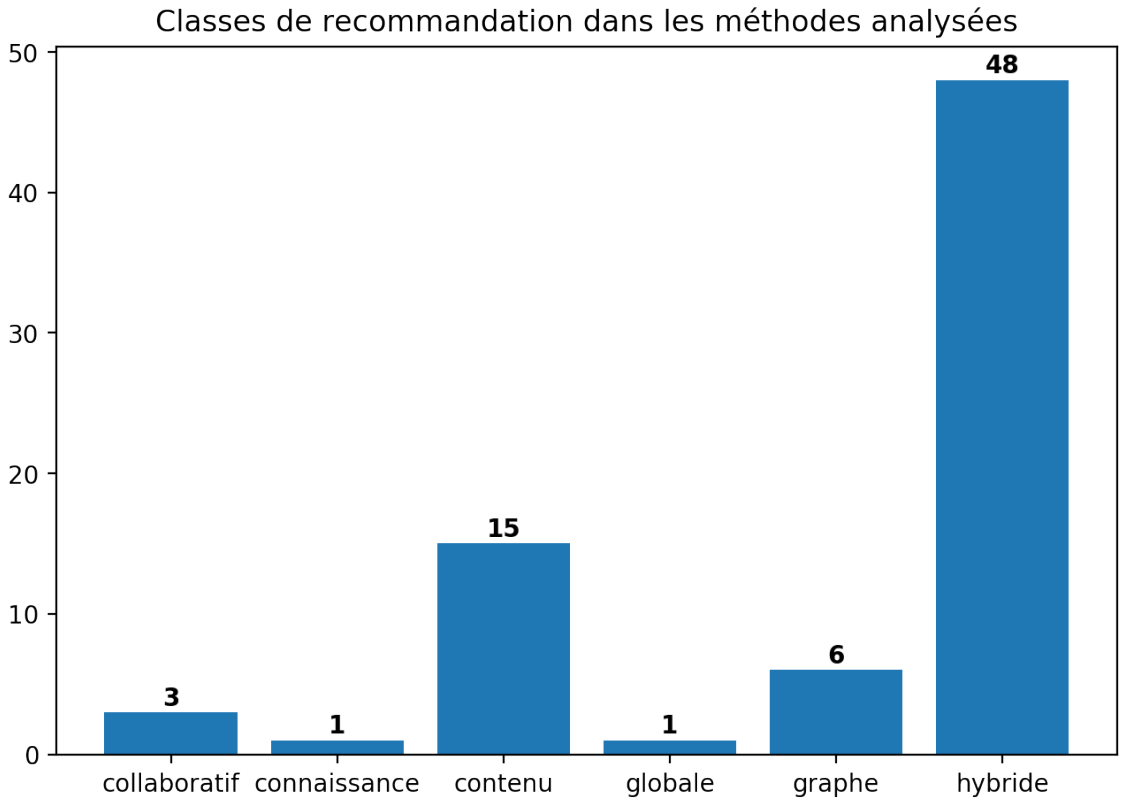

FIGURE 2.5 – Répartition des méthodes analysées dans les classes de recommandation (voir annexe A.3).

En conclusion, certaines tendances se dessinent dans la recherche comme on peut le voir sur la figure 2.5 qui montre la répartition des méthodes analysées dans les différentes classes de recommandation (voir annexe A.3). Celle-ci n'est pas le résultat d'une revue systématique mais permet tout de même quelques interprétations.

Tout d'abord, le filtrage collaboratif, qui est historiquement une des premières classes de recommandation et qui est encore largement employé aujourd'hui [de Gemmis et al., 2017], n'est pas beaucoup envisagé dans le domaine de la littérature scientifique à cause du manque de données. Ceci pourrait cependant évoluer avec le développement de plateforme de gestion de références bibliographiques (comme *Zotero*[6] ou *Mendeley*[7]), ou le développement des réseaux sociaux académiques (comme *Academia*[8]).

La recommandation basée sur le contenu, qui est historiquement la plus importante en littérature scientifique [Beel et al., 2016], reste bien présente. Elle possède en effet de nombreux avantages

6. https://www.zotero.org/
7. https://www.mendeley.com/
8. https://www.academia.edu/

comme la disponibilités des méta-données des articles et le développement des techniques de NLP. Enfin, il existe de nombreux outils et librairies qui facilitent le passage en production.

Mais force est de constater la prépondérance du filtrage hybride dans la recherche récente. La principale raison est sans doute la plus grande disponibilité des données tant pour la phase de conception (voir section 3.5) qu'en production (voir section 4.2). Il n'y a cependant pas de consensus quant à la meilleure approche aujourd'hui. Néanmoins, la combinaison de techniques basées sur les graphes et sur le contenu semble être la voie la plus prometteuse. Par contre, les méthodes hybrides sont complexes à mettre en œuvre. Et leur capacité à passer en production reste souvent problématique, alors qu'il s'agit pourtant d'un enjeu crucial [Beel et al., 2016].

2.3 Génération des recommandations

Dans la plupart des cas, il s'agit simplement de transmettre un classement des articles selon les prédictions d'évaluation obtenues lors de l'étape précédente [Ricci, 2017]. Cette liste peut être éventuellement tronquée pour des questions de lisibilité et/ou de performance. Ce qui s'avère d'ailleurs cohérent avec le but premier d'un système de recommandation qui est de filtrer l'information. Il s'agit du choix effectué dans la majorité des méthodes consultées.

Cependant, cette approche n'est pas toujours la meilleure. Par exemple, une liste trop restreinte des meilleurs scores risque de ne présenter à l'utilisateur que des items similaires et donc de manquer de diversité. De manière plus générale, la liste proposée doit être capable de répondre à différents besoins de l'utilisateur, qui sont parfois contradictoires [Ricci, 2017]. De plus, ces besoins sont parfois difficilement exprimables dans le système et surtout ils ne peuvent se réduire à une question de précision (i.e. de proximité sémantique).

2.4 Prise en compte du contexte

Cette section se base principalement sur Champiri et al. [2015] qui propose une revue de la littérature systématique des systèmes de recommandations de littérature scientifique prenant en compte le contexte.

Les classes, approches et méthodes présentées jusqu'à présent envisagent la recommandation comme un processus n'impliquant que des utilisateurs et des items. Cependant, un contexte peut être également associé à une évaluation. Il existe de nombreuses définitions du contexte dans le cadre de la recommandation [Champiri et al., 2015]. Par exemple, Lu et al. [2015] définit le contexte comme « toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité peut être une personne, un lieu ou un objet qui est considéré comme pertinent pour l'interaction entre un utilisateur et une application, ce qui inclus l'utilisateur et l'application eux-mêmes ». Celui-ci pourrait donc être exploité afin d'améliorer la personnalisation des recommandations, notamment en les faisant davantage correspondre aux besoins immédiats de l'utilisateur.

Formellement, la notion de transaction peut être redéfinie de la manière suivante [Ricci, 2017]. Soit $\mathcal{U}, \mathcal{I}, \mathcal{C}$, et $\mathcal{R}$ respectivement les ensembles des utilisateurs, des items, des contextes d'évaluation et des valeurs de l'échelle d'évaluation utilisée. La transaction entre un item i et un utilisateur u selon un contexte c est définie comme le quadruplet $(u, i, c, r(u, i, c)) \in \mathcal{U} \times \mathcal{I} \times \mathcal{C} \times \mathcal{R}$, avec $r : \mathcal{U}, \mathcal{I}, \mathcal{C} \rightarrow \mathcal{R}$ qui est la fonction d'évaluation.

Les systèmes de recommandations sensibles au contexte prennent donc en compte des aspects comme le temps, la localisation ou les caractéristiques de l'utilisateur. Champiri et al. [2015] pro-

pose d'ailleurs une liste plus systématique des informations pouvant potentiellement être prises en compte. L'intégration d'informations contextuelles dans le processus de recommandation peut être réalisée de multiples manières [Aggarwal, 2016, Champiri et al., 2015] (voir figure 2.6). Dans le cas où elle est réalisée avant ou après la prédiction de l'évaluation, elle a pour but principal de réduire l'ensemble des items candidats. Elle peut être également intégrée directement dans la prédiction des évaluations ou pour la génération des recommandations (comme critère de classement notamment).

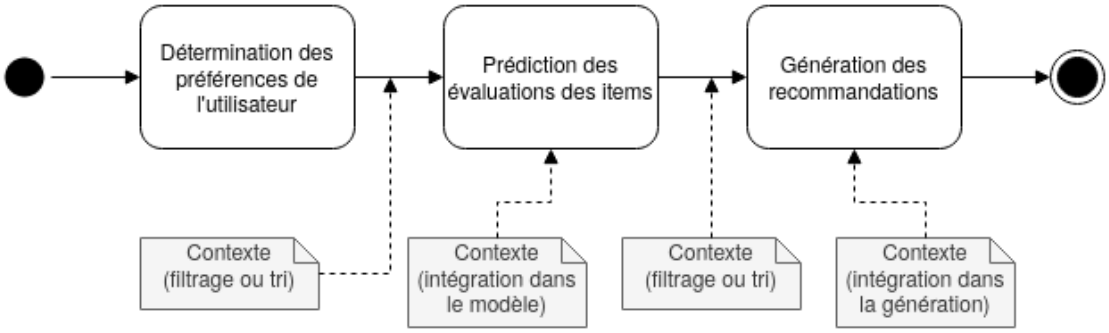

FIGURE 2.6 – Différentes manières d'intégrer le contexte au processus de recommandation, schéma inspiré de Champiri et al. [2015]

Dans la recommandation en littérature scientifique, le contexte n'est pas souvent pris en compte. Les quelques exemples rencontrés sont exclusivement consacrés à affiner les besoins de l'utilisateur. West et al. [2016] propose à l'utilisateur de choisir entre trois profils différents selon ses besoins (*expert*, *classique* et *sérendipité*). Ce choix détermine le nombre de candidats retenus et la fonction d'utilité appliquée. Zhao et al. [2016] considère les connaissances acquises par l'utilisateur (via l'historique de ses publications) et lui propose des recommandations permettant de faire le lien avec le sujet d'intérêt. Dhanda and Verma [2016] propose à l'utilisateur de choisir un critère de préférence (par exemple l'autorité ou la date) pour générer les recommandations. Enfin, Kobayashi et al. [2018] et Chakraborty et al. [2016] classent les citations selon différentes catégories sémantiques et permettent à l'utilisateur de choisir une catégorie à privilégier. À noter que ces deux dernières méthodes nécessitent l'accès aux textes intégraux des articles du corpus. Dans ces cinq méthodes, la prise en compte du contexte est faite par intégration dans le calcul des prédictions.

2.5 Recommandation multi-objectifs

Comme le souligne Adomavicius and Kwon [2015], le développement des méthodes de recommandation se concentre principalement sur l'amélioration de la précision des recommandations. Ce qui est également vrai dans le cas de la recommandation en littérature scientifique [Beel et al., 2016]. En effet, la majorité des méthodes ne prennent pas en compte le fait que la satisfaction des utilisateurs peut ne pas dépendre uniquement de la précision des recommandations, mais aussi d'autres aspects comme la diversité, la nouveauté ou encore la sérendipité.

Le champ de la recommandation multi-objectifs s'adresse donc à ce problème. Il existe plusieurs stratégies qui peuvent être adaptées à la recommandation de littérature scientifique. Les solutions mises en œuvre sont d'ailleurs fort proches de ce qui existe de le cadre de la prise en compte du contexte.

Une première stratégie est le reclassement des items après recommandation [Kaminskas and

Bridge, 2016]. Son intérêt principal est de pouvoir être intégrée facilement à un processus de recommandation existant. Plusieurs approches sont possibles :

- utiliser une approche gloutonne consistant à reclasser de manière incrémentale les items recommandés en maximisant une fonction objectif à chaque itération,
- considérer le reclassement comme un problème d'optimisation et utiliser des techniques associées,
- ou utiliser différentes fonctions de scoring permettant d'évaluer les items recommandés selon différents aspects et les combiner selon une pondération pouvant privilégier l'un ou l'autre aspect.

Une deuxième stratégie facilement intégrable est le filtrage d'items recommandés selon certains critères correspondant aux objectifs ciblés [Kotkov et al., 2016].

La troisième stratégie est de modifier une méthode de recommandation existante pour intégrer d'autres aspects [Kaminskas and Bridge, 2016]. L'intérêt majeure de cette stratégie est de pouvoir améliorer la scalabilité du processus de recommandation et d'éviter le surcoût du filtrage ou du reclassement. Elle est cependant plus complexe à mettre en œuvre. Un exemple d'application proposé par Rodriguez et al. [2012] consiste à partir d'une méthode existante et de la combiner avec d'autres caractéristiques via un nouveau modèle. L'intérêt de cette approche est de pouvoir facilement être intégrée à un système existant.

La quatrième stratégie est l'hybridation. Il s'agit de combiner différentes méthodes favorisant différents objectifs [Castells et al., 2015]. Ses avantages et inconvénients sont similaires à la troisième stratégie. Par exemple, Ribeiro et al. [2014] propose de s'appuyer sur le concept de frontière de Pareto. L'idée est de positionner les points d'intérêt dans un espace normé et de sélectionner les points qui ne sont pas dominés par d'autres (voir figure 2.7). Il propose une première forme d'hybridation de méthodes de recommandation via combinaison des résultats. Chaque item est caractérisé dans un espace à p dimensions, p étant le nombre de méthodes de recommandation employées, selon ses différents scores d'évaluation. Et cette première forme sélectionne les items appartenant à la frontière de Pareto. Il propose une seconde forme d'hybridation également par combinaison des résultats. L'idée ici est d'utiliser l'ensemble des poids pour les différentes méthodes employées qui favorise la présence sur la frontière de Pareto. L'espace est cette fois déterminé par les scores des différentes listes selon différents objectifs comme la précision, la diversité ou la nouveauté (voir figure 2.7).

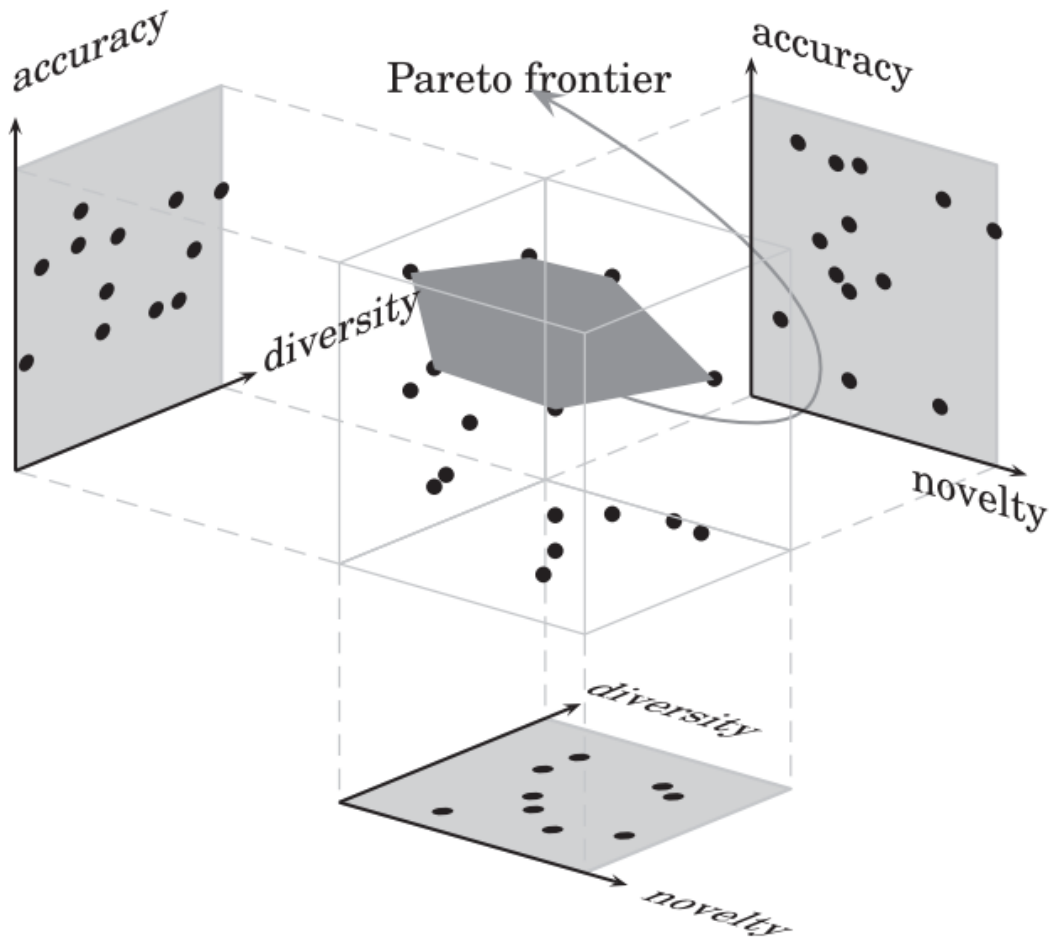

FIGURE 2.7 – Illustration de la frontière de Pareto dans un espace à 3 dimensions prenant en compte la nouveauté, la diversité et la précision. [Ribeiro et al., 2014]

Chapitre 3

Évaluation

Ce chapitre s'intéresse à l'évaluation des systèmes de recommandation. Il commence par définir la notion d'efficacité d'un système de recommandation et ses multiples aspects. Il présente ensuite les différents modes d'évaluation et techniques utilisés dans la littérature. Enfin, il s'intéresse aux jeux de données employés dans ce cadre.

3.1 Efficacité et évaluation

L'efficacité d'un système ou d'une méthode de recommandation est sa capacité à atteindre les objectifs fonctionnels et non-fonctionnels pour lesquels il a été conçu. Les principaux objectifs fonctionnels repris dans la littérature sont [Aggarwal, 2016, Beel et al., 2016, de Gemmis et al., 2017] :

- La **précision** des recommandations, c'est-à-dire la similarité sémantique par rapport au profil de l'utilisateur, et qui est l'objectif principal d'un système de recommandation.
- La **nouveauté** des recommandations pour l'utilisateur, avec comme but d'éviter les items plus anciens et/ou déjà connus.
- La **sérendipité** qui est la capacité des items recommandés à surprendre l'utilisateur.
- La **diversité** des recommandations afin de garantir qu'au moins un des items proposés intéressera l'utilisateur.
- En enfin la **couverture** du système de recommandation qui peut être envisagée selon les utilisateurs et les items.

Concernant les objectifs non-fonctionnels, les plus communs sont la scalabilité et la latence [Khusro et al., 2016], la robustesse et la stabilité [Aggarwal, 2016], ou encore les différents aspects liés à l'interaction entre l'utilisateur et le système de recommandation [Ricci et al., 2015]. Enfin, l'importance de la perspective de l'opérateur ne doit pas être négligée avec notamment des aspects comme les coûts de fonctionnement en production, la maintenabilité du système ou encore sa profitabilité [Beel et al., 2016].

L'évaluation permet d'estimer l'efficacité d'un système ou d'une méthode de recommandation. Les modes d'évaluation les plus courants sont [Beel and Langer, 2015] :

- Les **études utilisateurs** qui se focalisent sur la satisfaction des utilisateurs. Ces études peuvent être globales ou se concentrer sur des aspects spécifiques. Elles sont généralement employées pour valider un système de recommandation du point de vue de l'utilisateur finale et pour évaluer les aspects autres que la qualité des recommandations proprement dite

(comme l'expérience utilisateur par exemple) [Ricci et al., 2015].

— Les **évaluations en ligne** se focalisent plutôt sur l'évaluation des systèmes en production via les interactions avec des utilisateurs réels. Elles ont pour vocation de permettre une évaluation continue du système en production et de faciliter la mise en place d'améliorations d'un système existant [Ricci et al., 2015].

— Et enfin les **évaluations hors-ligne** sont plutôt utilisées en phase de conception afin d'évaluer la qualité des recommandations générées selon différents critères comme la précision, la nouveauté ou la sérendipité à partir de données historiques. Il s'agit du mode d'évaluation prédominant dans la littérature même s'il ne garantit pas toujours un système efficace en pratique [Beel et al., 2016].

Il existe de nombreux canevas permettant de définir un scénario d'expérimentation [Aggarwal, 2016, Gunawardana and Shani, 2015]. Trois étapes principales semblent toutefois revenir régulièrement. La première consiste en la définition des objectifs de l'évaluation, c'est-à-dire le ou les aspects à évaluer. La précision des recommandations est l'aspect le plus fréquemment envisagé mais il est important de ne pas se limiter à celui-ci pour améliorer la satisfaction des utilisateurs finaux [Aggarwal, 2016]. Il est également important que ces objectifs prennent la forme d'hypothèses concises et restrictives [Gunawardana and Shani, 2015] afin de faciliter la conception du protocole d'expérimentation.

La deuxième étape, qui est la conception du protocole d'expérimentation, comporte trois éléments importants. Tout d'abord, il est important de fixer les variables non testées lors de l'évaluation afin de garantir la juste interprétation des données obtenues [Gunawardana and Shani, 2015]. Ensuite, il faut choisir des alternatives pertinentes afin d'obtenir des résultats convaincants [Beel et al., 2016]. Par exemple, dans le cas de l'évaluation d'une nouvelle méthode de recommandation, il convient de choisir des méthodes représentatives de l'état de la littérature. Enfin, les mesures doivent être également choisies.

Et la troisième étape consiste en l'exploitation et l'interprétation des résultats. Outre le choix de visualisations appropriées, il est important de réaliser des tests statistiques adaptés aux données analysées afin de vérifier que les résultats sont bien significatifs [Gunawardana and Shani, 2015]. Ceux-ci appuieront notamment la puissance de généralisation de l'évaluation, c'est-à-dire le fait que les résultats obtenus ne sont pas spécifiques au contexte de réalisation de l'expérience [Gunawardana and Shani, 2015]. La question de la replicabilité et de la reproductibilité des expériences et des résultats est d'ailleurs un problème majeur de l'évaluation des systèmes de recommandation en littérature scientifique [Beel et al., 2016].

L'évaluation de l'efficacité dans la littérature se concentre trop sur la précision des recommandations au détriment des autres besoins de l'utilisateur, l'expérience utilisateur, l'IHM, etc. [Beel et al., 2016]. La majorité des méthodes ne prend pas en compte le fait que la satisfaction des utilisateurs peut ne pas dépendre uniquement de la précision des recommandations, mais aussi de la vie privée, de la sécurité des données, de la diversité, de la sérendipité, etc. Un élément lié est la négligence ou la faiblesse de la modélisation de l'utilisateur [Beel et al., 2016]. Celle-ci est rarement suffisamment aboutie et des raccourcis peu réalistes sont souvent pris.

3.2 Étude utilisateur

Une étude utilisateur consiste en l'observation et l'analyse d'utilisateurs interagissant avec le système de recommandation dans un environnement contrôlé ou non [Gunawardana and Shani, 2015].

Son principe repose sur le recrutement d'un panel d'utilisateurs potentiels. Ceux-ci effectuent plusieurs tâches avec le système de recommandation et durant lesquelles différents paramètres comme le temps, la précision, la réussite et la satisfaction sont collectés. Enfin, les utilisateurs peuvent également être soumis à un questionnaire en relation avec les tâches effectuées. Outre l'évaluation simple d'un système de recommandation unique, ce mode d'évaluation est également utilisé pour comparer plusieurs variantes d'un même système. Deux formules peuvent être employées dans ce cas : les utilisateurs peuvent être répartis entre les différentes variantes, ou les utilisateurs testent plusieurs variantes [Gunawardana and Shani, 2015].

Il s'agit du mode d'évaluation permettant de collecter le feedback le plus riche et le plus complet. Il permet de recueillir des données fines sur les interactions entre l'utilisateur et le système, en capturant notamment la perception subjective de la qualité du système [Bellogín and Said, 2017]. Sesagiri Raamkumar and Foo [2018] est un exemple de la richesse des objectifs et des aspects pouvant être captés par une étude utilisateur.

La technique de collecte la plus fréquemment utilisée dans la littérature, sans doute parce qu'elle est la plus simple à mettre en œuvre, est le questionnaire. Afin de simplifier l'exploitation des résultats, le questionnaire est en général constitué d'affirmations que l'utilisateur doit évaluer selon une échelle de satisfaction (souvent une échelle de Likert) [Sesagiri Raamkumar et al., 2017]. Un autre aspect spécifique aux systèmes de recommandation est l'évaluation de la pertinence des recommandations proposées, soit via une échelle binaire Zhao et al. [2016], soit via une échelle de satisfaction [Kanakia et al., 2019]. Enfin, du feedback libre peut être également recueilli [Sesagiri Raamkumar et al., 2017].

Dans tous les cas, la bonne conception du questionnaire est essentielle afin d'obtenir des résultats exploitables [Beel et al., 2016]. Outre le fait d'éviter les questionnaires trop fastidieux à remplir, il s'agit surtout de ne pas influencer l'utilisateur dans la rédaction des questions pour ne pas induire de biais dans l'évaluation. De manière générale, le fait de savoir qu'il participe à un test peut tout de même biaiser le comportement et le feedback de l'utilisateur [Gunawardana and Shani, 2015]. À noter l'existence de certains frameworks comme Knijnenburg and Willemsen [2015] et Pu et al. [2011] qui facilitent la conception.

L'étude utilisateur est aussi le mode d'évaluation le plus difficile à mettre en œuvre, notamment à cause du temps nécessaire à sa conduite, allant de quelques semaines [De Nart and Tasso, 2014] à plusieurs mois [Sesagiri Raamkumar et al., 2017], mais également parce qu'il faut un groupe suffisamment nombreux de personnes, idéalement représentatif des utilisateurs finaux [Gunawardana and Shani, 2015]. Il s'agit donc d'un mode d'évaluation peu employé dans la littérature. Qui plus est, peu de méthodes réalisant une étude utilisateur ont un nombre suffisant de participants pour pouvoir tirer des conclusions significatives [Beel et al., 2016]. Knijnenburg and Willemsen [2015] et Demšar [2006] approfondissent les questions liées à la construction d'un panel d'utilisateurs.

Une alternative à l'étude utilisateur (avec un panel représentatif et suffisamment fourni) est l'étude pilote [Gunawardana and Shani, 2015]. Elle est moins ambitieuse en terme de résultats mais elle peut être utile pour valider un système, identifier les bugs et d'autres problèmes de fonctionnement en préalable à une étude utilisateur de plus grande ampleur.

En conclusion, l'étude utilisateur est véritablement le meilleur moyen d'évaluer un système de recommandation mais également le plus complexe à mettre en œuvre.

3.3 Évaluation en ligne

L'évaluation en ligne consiste en l'analyse des interactions entre utilisateurs et un système de recommandation par l'intermédiaire des traces laissées par ces derniers (i.e. les logs). Celles-ci permettent de reconstruire différents éléments comme les actions effectuées, le temps passé sur les différentes pages, etc. Ce mode d'évaluation nécessite généralement de disposer d'un système de recommandation déjà déployé en production, ce qui explique sans doute pourquoi il est peu employé dans la littérature [Beel et al., 2016]. Sa motivation principale est de monitorer un système dans un contexte de production et avec de véritables utilisateurs, qui n'ont généralement pas conscience d'être observés, afin de comprendre l'influence des différents composants sur ceux-ci [Gunawardana and Shani, 2015]. De plus, l'évaluation en ligne permet d'étudier de nombreux aspects du système comme les méthodes de recommandation employées, mais également différents éléments de l'interface, la présentation des résultats, etc. Il est d'ailleurs important de bien définir ce qui est évalué et de fixer les autres paramètres afin d'obtenir des données exploitables [Gunawardana and Shani, 2015].

L'approche généralement employée pour sélectionner les utilisateurs est l'A/B testing. Elle consiste à rediriger, à leur insu, des utilisateurs choisis aléatoirement vers une version alternative du système de recommandation où le paramètre étudié est modifié [Gunawardana and Shani, 2015]. Cette approche peut même déboucher sur une automatisation de l'exploitation des résultats de manière similaire au problème du bandit manchot [Aggarwal, 2016].

Ce mode d'évaluation peut être utilisé avec des mesures de précision mais il donne généralement de meilleurs résultats avec des mesures spécifiques. La mesure la plus fréquemment utilisée est le taux de clics, ou *Click-Through Rate* (CTR) [Beel et al., 2016], qui est le rapport entre le nombre d'items consultés et le nombre d'items présentés. Cette mesure, originaire de la publicité en ligne et de l'e-commerce, repose sur la supposition que la consultation implique un intérêt et donc une évaluation positive. Néanmoins, il n'y a pas toujours corrélation entre CTR et pertinence des recommandations [Beel et al., 2016]. D'autres mesures comme le taux de téléchargement ou le taux d'achat peuvent être également employées. En général, il y a une bonne corrélation entre les résultats obtenus via ce mode d'évaluation et les études utilisateurs [Beel and Langer, 2015].

Comme cela a été dit précédemment, l'évaluation en ligne est peu employée dans la littérature principalement car elle nécessite l'accès à un système de recommandation en production. Cependant, Beel, Collins, Kopp, Dietz and Knoth [2019] a lancé l'initiative *Mr. DLib*[1] qui est une plateforme de recommandation en partenariat avec le gestionnaire de références *JabRef*[2] et la société CORE[3]. Cette plateforme a pour but principal de mettre à disposition de divers acteurs académiques un système de recommandation sous forme d'API afin que ces derniers puissent enrichir les services proposés à leurs utilisateurs. Elle propose également un service d'évaluation en ligne sous forme d'une API permettant de de brancher un moteur de recommandation externe et de récupérer des logs concernant les interactions des utilisateurs avec ce moteur (clics, téléchargements, achats, etc.). Beel, Dinesh, Mayr, Carevic and Raghvendra [2017] est un exemple de méthode exploitant cette plateforme. Une autre possibilité est l'environnement de test proposé par *Social Science Research Network*[4] et notamment exploité par West et al. [2016].

1. `http://mr-dlib.org/`
2. `https://www.jabref.org/`
3. `https://core.ac.uk/` active dans le domaine de la documentation scientifique
4. `https://www.ssrn.com/`

3.4 Évaluation hors-ligne

L'évaluation hors-ligne a pour but d'estimer différents aspects de l'efficacité d'une méthode ou d'un système de recommandation avant sa mise en production. Des corpus d'articles fermés et bien définis sont utilisés la plupart du temps afin de garantir la réplicabilité des expériences. Il peut s'agir d'articles collectées spécifiquement par l'équipe de développement ou de corpus mis à disposition par des producteurs de données bibliographiques. L'utilisation de données historiques liées aux transactions entre utilisateurs et items à la place d'utilisateurs réels facilite énormément la mise en œuvre des expérimentations [Gunawardana and Shani, 2015]. Ce mode d'évaluation est d'ailleurs fréquemment utilisé en début de conception d'un système de recommandation afin de sélectionner les méthodes les plus prometteuses.

Cependant, plusieurs recherches montrent que pour les systèmes de recommandation en général, les résultats des évaluations hors-ligne ne sont pas nécessairement corrélés avec les résultats des études utilisateurs et des évaluations en ligne [Aggarwal, 2016, Bellogín and Said, 2017]. C'est également le cas pour les systèmes spécifiques à la littérature scientifique [Beel and Langer, 2015]. La reproductibilité des évaluations hors-ligne est également problématique [Beel et al., 2016]. D'importantes variations entre les résultats peuvent par exemple être constatées en fonction des jeux de données utilisés. De manière générale, il convient d'être conscient des biais induits par les méthodes d'évaluation hors-ligne, notamment liés au choix des jeux de données et des mesures qui peut conduire à sur/sous-estimer les performances d'une méthode ou d'un système [Aggarwal, 2016]. Enfin, ce mode d'évaluation reste limité à la comparaison de méthodes de recommandation et ne permet pas d'évaluer d'autres aspects d'un système de recommandation [Gunawardana and Shani, 2015].

3.4.1 Précision : évaluation supervisée

La plupart des évaluations hors-ligne rencontrées dans la littérature ont pour but l'évaluation de la précision des recommandations, c'est-à-dire la similarité sémantique des items recommandés avec le sujet d'intérêt de l'utilisateur. L'hypothèse générale qui sous-tend l'évaluation supervisée de la précision est qu'une méthode efficace sur un historique de transactions entre utilisateurs et items sera également efficace avec les utilisateurs finaux en production [Bellogín and Said, 2017].

Jusqu'à récemment, de telles données étaient cependant difficiles à obtenir dans le cadre de la recommandation en littérature scientifique [Bai et al., 2019]. La majorité des protocoles d'évaluation hors-ligne s'appuie donc généralement sur les relations de citations entre articles comme substituts aux évaluations. Plusieurs interprétations de ces relations sont présentes dans la littérature. La plus répandue consiste à considérer un article comme étant un utilisateur et ses références bibliographiques comme étant des articles pour lesquels l'utilisateur émet une évaluation positive. L'échelle d'évaluation utilisée est donc une échelle unaire (i.e. évaluation positive ou pas d'évaluation). Une explication du succès de cette interprétation est sans doute la proximité avec le contexte de la rédaction d'un article scientifique qui est un cas d'utilisation assez fréquent.

Une autre interprétation rencontrée dans la littérature est d'utiliser les relations d'auteurs entre articles et utilisateurs [Dai, Gao, Zhu, Cai and Pan, 2018]. Chaque utilisateur est donc caractérisé par les articles dont il est auteur, également selon une échelle unaire. Le problème principal de cette interprétation, qui explique sans doute son faible succès, est qu'un chercheur peut potentiellement être auteur d'articles sur des sujets différents et l'ensemble de ses publications risque donc d'être

trop hétérogène pour avoir du sens dans le cadre d'une recommandation. Une dernière interprétation rencontrée dans la littérature est d'utiliser les articles cités par les auteurs dans leurs publications [Amami et al., 2016, Chen and Lee, 2018]. Chaque utilisateur est donc associé aux articles qu'il a cités dans ses différentes publications. Même si cette interprétation souffre du même problème que la précédente, elle a l'avantage de permettre l'utilisation d'une échelle numérique en prenant en compte le nombre de fois où chaque article est cité, et donc de disposer d'évaluations plus nuancées.

Sur le plan pratique, la séparation entre données d'entraînement et données de test est évidemment primordiale afin de garantir des résultats valables [Bellogín and Said, 2017]. Différentes techniques comme la séparation simple des données utilisées (entre jeux d'entraînement, de test et éventuellement de validation) ou des techniques plus avancées comme la validation croisée sont employées dans la littérature. Dans le cas d'une séparation simple, l'utilisation d'une date pivot et la séparation aléatoire sont les méthodes les plus fréquemment utilisées.

Enfin, il est important de remarquer que l'utilisation des citations ou des relations d'auteurs comme substituts à de véritables données d'évaluation pose un réel problème lorsqu'il s'agit de comparer des classes différentes de recommandation. En effet, certaines méthodes utilisent ces données pour générer des recommandations et sont donc favorisées par rapport à des méthodes qui ne les utilisent pas. Cependant, l'apparition de certains jeux de données comme RARD (*Related-Article Recommendation Dataset*) [Beel, Carevic, Schaible and Neusch, 2017, Beel, Smyth and Collins, 2019] ou *CiteULike* [Wang et al., 2013] avec de véritables données d'évaluation pourrait bien permettre de contourner ce problème.

Précision des prédictions

Le premier ensemble de mesures regroupe les mesures qui comparent les évaluations prédites aux évaluations réelles [Bobadilla et al., 2013]. L'hypothèse de départ est de considérer qu'un bon système de recommandation est capable de produire des prédictions proche de la réalité [Bellogín and Said, 2017]. L'erreur moyenne absolue ou *mean absolute error* (MAE) et l'erreur moyenne quadratique ou *root mean square error* (RMSE) sont les mesures plus utilisées [Bai et al., 2019]. RMSE permet de connaître l'erreur moyenne entre les évaluations réelles des items par les utilisateurs et les prédictions par le système de recommandation :

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}$$

où n est le nombre d'évaluations, r_i est l'évaluation réelle et $\hat{r}_i$ est l'estimation. Et MAE est définie comme

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|r_i - \hat{r}_i|$$

À noter que RMSE met davantage d'emphase sur les erreurs importantes que MAE.

Ces mesures sont rarement employées dans le cadre des systèmes de recommandation de littérature scientifique étant donné la difficulté d'obtenir des évaluations [Bai et al., 2019]. De plus, elles ne correspondent pas véritablement à l'objectif d'un système de recommandation qui est de générer une liste d'items susceptibles d'intéresser l'utilisateur et non de produire les prédictions les plus proches des évaluations réelles [Bellogín and Said, 2017].

Précision des ensembles de recommandations

Le deuxième ensemble de mesures s'intéresse au nombre d'items corrects (i.e. qui intéressent effectivement l'utilisateur) parmi les items recommandés [Bobadilla et al., 2013]. Les principales mesures utilisées sont la précision, le rappel, le F-score et les différentes courbes associées (ROC, etc.) [Bai et al., 2019, Isinkaye et al., 2015]. Ces mesures sont davantage employées dans la littérature d'une part parce qu'elles ne nécessitent que des scores d'évaluation de type unaire ou binaire, et d'autre part parce qu'elles évaluent la qualité de l'ensemble des recommandations, ce qui est plus proche des cas d'utilisation ciblés.

La précision évalue le nombre d'items correctement recommandés par rapport nombre total d'items recommandés.

$$Precision = \frac{items\ pertinents\ recommandés}{items\ recommandés}$$

Il existe également une version de complexité moindre limitant la taille de l'ensemble d'items pris en compte. Le rappel évalue le nombre d'items correctement recommandés par rapport au nombre total d'items corrects.

$$Rappel = \frac{items\ pertinents\ recommandés}{items\ pertinents}$$

Comme pour la précision, il existe également une version prenant en compte un nombre limité d'items. Et le F-score qui est la moyenne harmonique du rappel et de la précision.

$$F = \frac{(\alpha^2 + 1)(Precision \times Rappel)}{\alpha^2(Precision + Rappel)}$$

avec le paramètre α qui permet de privilégier le rappel ou la précision en fonction de sa valeur. Une valeur de 1 est la plus fréquemment employée, ce qui permet de considérer le rappel et la précision d'égale importance.

Le principal défaut de ce deuxième ensemble est de ne pas prendre en compte l'ordre des items. Un item incorrect en début de classement est pénalisé de la même manière qu'un item incorrect en fin de classement [Bellogín and Said, 2017], or l'erreur ne sera pas perçue de la même manière du point de vue de l'utilisateur.

Précision des listes classées de recommandations

Le dernier ensemble de mesures s'intéresse au classement produit [Bobadilla et al., 2013]. L'hypothèse est qu'une bonne méthode de recommandation ne doit pas seulement maximiser le nombre d'items corrects, mais les items corrects doivent être mieux classés que les items incorrects. Il s'agit sans doute de l'hypothèse la plus proche de la réalité. Il est en effet admis que les listes de résultats obtenus par un système de recommandation ne sont guère consultées au delà d'un certain nombre de résultats (à l'image des résultats d'un moteur de recherche classique). L'évaluation de la précision du classement est d'ailleurs de plus en plus privilégiée dans le domaine [Bellogín and Said, 2017].

Une première mesure sensible à l'ordre des items est la moyenne des précisions moyennes ou *mean average precision* (MAP) [Bai et al., 2019]. La précision moyenne pour une liste de recommandations est définie comme

$$AP = \frac{1}{m} \sum_{k=1}^{n} P(R_k)$$

Pour une liste de recommandations R, m est le nombre d'items corrects, n est le nombre d'items recommandés, et $P(R_k)$ représente la précision de l'ensemble R_k contenant les k premiers items

recommandés. Et la moyenne des précisions moyennes est donc définie comme

$$MAP = \frac{1}{m} \sum_{k=1}^{m} AP_k$$

m étant le nombre de listes de recommandations et AP_k la précision moyenne pour la liste k.

Une autre mesure spécifique et le rang réciproque moyen ou *mean reciprocal rank* (MRR) [Bai et al., 2019] qui s'intéresse au classement moyen du premier item pertinent dans une liste d'items recommandés. Elle est définie comme

$$MRR = \frac{1}{m} \sum_{k=1}^{m} \frac{1}{rank_k}$$

où m est le nombre de listes de recommandations et $rank_k$ représente le rang du premier item pertinent dans la liste k.

Mais la mesure la plus fréquemment employée est le gain cumulatif discontinu ou *discounted cumulative gain* (DCG) [Bai et al., 2019]. Il est également utilisé pour évaluer la qualité d'une liste triée d'items recommandés. Il est défini comme

$$DCG = \frac{1}{m} \sum_{k=1}^{m} \sum_{i=1}^{n} \frac{g_{ki}}{max(1, \log_b(i))}$$

où m est le nombre de listes de recommandations, n est le nombre d'items recommandés pour une liste, i est la position de l'item dans la liste, b est une valeur constante, g_{ki} représente le gain obtenu avec l'item i dans la liste k. Il peut être utilisé tel quel ou dans sa version normalisée ($nDCG = \frac{DCG}{max(DCG)}$). L'intérêt majeur de cette mesure est qu'elle peut être utilisée des évaluations non binaires ou intégrer une fonction d'utilité [Aggarwal, 2016].

3.4.2 Précision : évaluation non supervisée

Cependant, l'évaluation supervisée de la précision est problématique pour deux raisons. D'une part, le fait d'utiliser la bibliographie des articles comme liste de référence implique de considérer celle-ci comme représentative du sujet d'intérêt (i.e. des articles en entrée). Alors qu'une bibliographie est plutôt une sélection partielle et subjective, et où certaines références peuvent notamment être substituées par d'autres. Le risque principal est donc de sous-évaluer des méthodes recommandant des articles pertinents mais qui n'appartiennent pas à la bibliographie. D'autre part, elle avantage indubitablement les méthodes utilisant le graphe des citations au détriment des autres (comme celles basées sur le contenu par exemple), ce qui pose problème dans le cas où des méthodes appartenant à différentes classes de recommandation sont comparées.

Une autre approche, inspirée de ce qui est réalisé en apprentissage non-supervisé, consiste à évaluer les qualités intrinsèques des listes de recommandations. Dans ce cas, la précision est ici formulée comme la similarité sémantique des items recommandés avec le profil de l'utilisateur cible (i.e. le ou les articles d'intérêt). Dans le cas de la recommandation de littérature scientifique, la similarité sémantique peut être envisagée principalement du point de vue textuel en comparant les titres et les abstracts, via l'utilisation d'une base de connaissance comme un ensemble de mots-clés, et du point de vue topologique en comparant les références et les citations. D'autres aspects peuvent aussi être envisagés comme les auteurs ou les organes de publication.

Similarité textuelle

L'évaluation de la similarité textuelle consiste à comparer les contenus textuels des documents pour déterminer leur similarité. Dans le cas d'articles, il s'agit le plus souvent des titres et des abstracts. La similarité textuelle comporte deux aspects : le choix de la représentation et la fonction de comparaison. Le choix de la représentation renvoie vers des techniques en provenance du NLP et déjà évoquées dans la recommandation basée sur le contenu (voir section 2.2.3) : sacs-de-mots, *word embeddings*, extraction de sujets, etc. La fonction de comparaison la plus fréquente est la similarité cosinus. Il existe néanmoins des alternatives comme *word mover's distance* [Kusner et al., 2015] qui est plus performante mais également plus coûteuse en calcul.

Il s'agit sans doute de l'approche la plus intuitive pour évaluer la similarité sémantique entre articles. De plus, les titres et les abstracts sont des méta-données disponibles la plupart du temps. Cependant, l'inconvénient majeur de la similarité textuelle est qu'elle risque de favoriser les méthodes de recommandation basées sur le contenu. Car les techniques évoquées ci-dessus sont souvent employées dans cette classe.

Similarité à partir d'une représentation des connaissances

Une représentation des connaissances est une représentation formelle d'un domaine de connaissance constituée de concepts organisés selon des relations sémantiquement signifiantes. Il existe différents types de représentation [Gilchrist, 2003] :

- la taxonomie (ou système de classification) est une organisation hiérarchique des connaissances plutôt utilisée dans des tâches de classification documentaire,
- le thésaurus est un ensemble de concepts reliés par des relations hiérarchiques ou d'association et plutôt utilisé en indexation documentaire,
- et l'ontologie est une modélisation d'un domaine de connaissance sous forme de concepts et de relations permettant de « raisonner » sur celui-ci.

Des exemples de représentations des connaissances utilisés pour l'indexation d'articles sont : le MeSH (*Medical Subject Headings*) [5], les *Fields of Study* (*Microsoft Academic*) [6] et le système de classification d'ACM [7].

L'utilisation d'une représentation des connaissances comporte plusieurs avantages. Tout d'abord, les représentations formelles des connaissances s'appuient sur un langage contrôlé, ce qui permet d'éviter les problèmes du langage naturel comme la polysémie, les synonymes, les variations terminologiques, etc. Comparativement à une indexation via des mots-clés librement ajoutés par les auteurs, une indexation via base de connaissance est globalement plus homogène.

Ensuite, les différents concepts sont associés selon des relations sémantiquement signifiantes (souvent une hiérarchie), ce qui permet une comparaison fine prenant en compte ces relations. Enfin, l'usage de mots-clés par les méthodes de recommandation en littérature scientifique, et à plus forte raison de mots-clés provenant d'une base de connaissance, est beaucoup moins fréquent. Ce qui limite les risques de biais dans le cas de comparaison de nombreuses méthodes. Une explication de cet usage moins fréquent est sans doute que l'utilisation de bases de connaissance pour l'indexation est lié à des sources de données bien spécifiques. Ce qui est difficilement transposable, notamment lorsque le corpus d'items en production est constitué à partir de plusieurs sources de données.

5. `https://www.nlm.nih.gov/mesh/meshhome.html`
6. `https://academic.microsoft.com/topics`
7. `https://www.acm.org/publications/class-2012`

Par contre, cette approche possède également quelques inconvénients. Tout d'abord, elle nécessite une méthode de comparaison prenant en compte les spécificités de la représentation des connaissances employée. Et il existe peu de jeux de données disponibles s'appuyant sur une représentation des connaissances pour l'indexation.

Cette approche n'est malheureusement pas vraiment employée en recommandation de littérature scientifique. La seule initiative identifiée est le framework d'évaluation CITREC [Gipp and Meuschke, 2015]. Celui-ci propose notamment deux jeux de données constitués d'articles dans le domaine médical. Il contient également une série de mesures de similarité dont une basée sur l'utilisation du MeSH. Un exemple d'utilisation de ce framework est proposé par Tian and Zhuo [2017].

Similarité topologique (i.e. références et citations similaires)

La similarité entre deux articles peut également s'envisager du point de vue de leurs références (i.e. leurs bibliographies) et citations (i.e. les citations par d'autres articles). L'idée étant bien sûr qu'un nombre important de références et de citations similaires induit une similarité sémantique.

La première mesure de similarité, et sans doute la plus répandue, est le couplage bibliographique [Kessler, 1963]. L'idée est de considérer deux articles comme similaires si leurs bibliographies partagent de nombreuses références communes. Formellement, celle-ci est définie comme

$$\textit{couplage bibliographique} = \frac{|R_1 \cap R_2|}{|R_1 \cup R_2|}$$

avec R_1 et R_2 comme les ensembles des références bibliographiques des articles comparés.

Une deuxième mesure fréquemment employée est la co-citation [Small, 1973]. Elle considère la similarité entre deux articles comme étant liée au nombre citations communes, selon la formule

$$\textit{co-citation} = \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$$

avec C_1 et C_2 comme ensembles des citations des articles comparés. Bien qu'analogue au couplage bibliographique, la co-citation semble moins efficace pour évaluer la similarité sémantique entre articles selon diverses études [Boyack and Klavans, 2010, Couto et al., 2006, Sternitzke and Bergmann, 2009], tant pour des tâches de classification que de clustering. Une explication de cette moindre précision pourrait être la plus grande variabilité dans le nombre de citations des articles des corpus utilisés (phénomène constaté dans ce travail). Par ailleurs, Steinert and Hoppe [2016] constate empiriquement l'absence de corrélation entre les mesures de couplage bibliographique et de co-citation, avec pour conséquence des classements obtenus fort différents.

Enfin, une troisième mesure de similarité dite de Amsler [Amsler, 1972] prend en compte à la fois les références et les citations de la manière suivante :

$$\textit{similarité de Amsler} = \frac{|(R_1 \cup C_1) \cap (R_2 \cup C_2)|}{|(R_1 \cup C_1) \cup (R_2 \cup C_2)|}$$

Cette mesure équivaut à l'indice de Jaccard calculé sur le graphe non-orienté des citations entre articles. Elle est nettement moins connue et employée que le couplage bibliographique et la co-citation. Cependant, Couto et al. [2006] montre pour des tâches de classification la meilleure performance de Amsler, suivi de près par le couplage bibliographique, les deux dominant largement la co-citation. De plus, le fait que la similarité de Amsler exploite à la fois les références et les citations offre un intérêt pratique qui est de mieux exploiter les jeux de données avec des citations incomplètes, ce qui

est régulièrement le cas étant donné que ces données sont souvent difficiles à extraire (voir section 3.5).

Outre ces mesures issues de la scientométrie, les techniques d'*embeddings* appliquées au graphe des citations peuvent également être employées. L'approche est alors similaire à la similarité textuelle évoquée ci-dessus : le choix d'une technique de représentation (par exemple DeepWalk ou Node2Vec) et d'une fonction de similarité (par exemple la similarité cosinus).

Outre les critiques déjà émises par rapport à l'utilisation des citations pour l'évaluation supervisée de la précision, il semble que la similarité topologique soit moins performante que la similarité textuelle pour évaluer la similarité sémantique entre deux documents [Bhattacharya et al., 2020]. De plus, les citations ne sont pas toujours disponibles. Ces raison expliquent sans doute que la similarité topologique sur le graphe des citations ne soit pas vraiment employée pour évaluer les méthodes de recommandation en littérature scientifique.

3.4.3 Diversité

La diversité d'une liste de recommandations est une propriété qui caractérise la variété des items proposés, c'est-à-dire le fait qu'ils soient plus ou moins différents les uns par rapport aux autres. L'objectif derrière la diversité est de proposer une sélection suffisamment large d'items afin de couvrir largement le sujet d'intérêt et, *in fine*, favoriser la satisfaction de l'utilisateur [Kaminskas and Bridge, 2016]. Il s'agit notamment d'apporter une réponse à une requête qui n'est pas toujours bien définie ou qui comporte des ambiguïtés. Ce qui est d'autant plus vrai étant donné la multiplicité des approches possibles par rapport à un sujet donné en littérature scientifique.

L'approche la plus fréquemment employée pour estimer la diversité d'une liste consiste à calculer la dissimilarité (ou distance) moyenne entre les items de cette liste [Kaminskas and Bridge, 2016] :

$$diversité = \frac{\sum_{i \in R} \sum_{j \in R \setminus \{i\}} dist(i,j)}{|R|(|R|-1)}$$

R étant un ensemble d'items recommandés et $dist(i,j)$ une fonction de distance entre deux items i et j. Différentes fonctions de distance peuvent être employées en fonction du domaine de recommandation. Deux fonctions fréquentes sont le complément de l'indice de Jaccard et le complément de la similarité cosinus [Kaminskas and Bridge, 2016]. Kaminskas and Bridge [2016] pointe également quelques critiques envers cette approche et évoque quelques alternatives.

3.4.4 Nouveauté

La nouveauté d'une liste de recommandations caractérise le fait que celle-ci contienne un certain nombre d'items inconnus de l'utilisateur, tout en restant proches du sujet d'intérêt [Kaminskas and Bridge, 2016]. L'approche d'évaluation la plus intuitive pour estimer la nouveauté est évidemment de prendre en compte la nouveauté temporelle. Dans le domaine de la littérature scientifique, une mesure peut être par exemple la date de publication moyenne d'une liste d'items recommandés. Cette approche fait d'ailleurs particulièrement sens dans ce domaine étant donné l'importance de l'actualité de la recherche. Elle ne considère cependant pas directement la nouveauté du point de vue de l'utilisateur.

Une seconde approche consiste à partir du constat que les items peu populaires ont plus de chance d'être inconnus d'un utilisateur donné que les items populaires [Kaminskas and Bridge, 2016]. L'idée est donc d'utiliser le nombre d'évaluations des items comme un indicateur de cette

popularité. La nouveauté peut alors être définie comme :

$$nouveauté = \frac{\sum_{i \in R} -log_2 p(i)}{|R|}$$

$1 - p(i)$ étant le complément de la popularité définie comme

$$p(i) = \frac{|\{u \in U, r_{ui} \neq \emptyset\}|}{|U|}$$

avec U l'ensemble des utilisateurs du système et r_{ui} l'évaluation de l'item i par l'utilisateur u. Dans le cas de la littérature scientifique, le nombre de citations peut être une approximation de la popularité, bien que celle-ci soit sujette à caution étant donné qu'un faible nombre de citations peut également être le signe d'un travail de moindre qualité.

Enfin, il est également possible d'évaluer la nouveauté de manière supervisée. L'idée est alors de considérer pour chaque utilisateur une date charnière. Cette date permet de définir pour chaque utilisateur un ensemble d'items évalués après cette date. Et la nouveauté est estimée via le nombre d'items recommandés appartenant à cet ensemble [Gunawardana and Shani, 2015].

3.4.5 Sérendipité

La notion de sérendipité n'est pas évidente à définir comme l'illustre l'absence de consensus dans la littérature [Kotkov et al., 2016]. Néanmoins, la plupart des auteurs s'accordent pour dire que la sérendipité consiste à proposer des recommandations inattendues et qui vont surprendre l'utilisateur [Gunawardana and Shani, 2015]. Kaminskas and Bridge [2016] pointe également la proximité de la sérendipité avec la nouveauté pour l'utilisateur, tout en distinguant la notion par l'effet de surprise et l'importance de maintenir un certain niveau de précision par rapport au sujet d'intérêt.

La première approche pour estimer la sérendipité consiste à cerner cette notion par décomposition de ces différentes composantes : la nouveauté, l'inattendu et la précision [Kotkov et al., 2016]. La nouveauté et la précision peuvent être estimées via les mesures évoquées ci-dessus. Et le caractère inattendu des recommandations peut être estimé par le biais de la dissimilarité avec le profil de l'utilisateur [Kotkov et al., 2016]. À première vue, cette approche de l'estimation du caractère inattendu semble contradictoire avec la précision. Néanmoins, l'utilisation de critères de similarité différents peut être une piste intéressante. Par exemple, un critère de similarité textuelle pourrait être utilisé pour la précision et un critère de dissimilarité topologique pour l'inattendu.

La seconde approche considère la sérendipité globalement et l'évalue par comparaison avec une méthode de recommandation ne favorisant pas l'effet de surprise [Kaminskas and Bridge, 2016]. Elle se base sur la définition suivante de la sérendipité :

$$sérendipité = \frac{|R_{inattendu} \cap R_{utile}|}{|R|}$$

R étant un ensemble de recommandation pour un utilisateur u, $R_{inattendu}$ étant l'ensemble des items inattendus et R_{utile} l'ensemble des items utiles. L'ensemble des items inattendus est alors défini comme

$$R_{inattendu} = R \setminus PM_u$$

PM_u étant l'ensemble des items recommandés par une méthode ne favorisant pas la sérendipité.

3.4.6 Couverture

La notion de couverture peut concerner les utilisateurs et les items. Dans le cas des utilisateurs, la couverture est définie comme le taux d'utilisateurs $U_{pertinent}$ pouvant obtenir des recommandations pertinentes parmi les utilisateurs potentiels U [Bai et al., 2019] :

$$couverture\ utilisateurs = \frac{|U_{pertinent}|}{|U|}$$

L'intérêt direct de cette mesure est évident pour un système en production étant donné son impact sur la satisfaction utilisateur.

Dans le cas des items, la couverture est définie comme le taux d'items effectivement recommandés parmi les items du corpus [Kaminskas and Bridge, 2016]

$$couverture\ items = \frac{|\bigcup_{u \in U} R_u|}{|I|}$$

où R_u est l'ensemble des recommandations pour l'utilisateur u appartenant à l'ensemble des utilisateurs U, et I est l'ensemble des items du corpus. L'intérêt d'une bonne couverture d'items est de favoriser les qualités de diversité, de sérendipité et de nouveauté des recommandations. Ces qualités sont en effet liées d'une certaine manière à la richesse de l'ensemble des items recommandables. Bien que la relation soit reconnue dans la littérature, elle peut ne pas toujours être évidente [Kaminskas and Bridge, 2016]. Il existe certaines variantes plus élaborées, prenant par exemple en compte la distribution des fréquences des items recommandés auprès des utilisateurs [Kaminskas and Bridge, 2016].

3.5 Jeux de données

Les jeux de données (ou *datasets*) sont des corpus de données fermés dont la vocation première est de pouvoir comparer différentes méthodes de recommandation. Le caractère fermé des jeux de données est essentiel afin de garantir que les méthodes soient comparées sur un même ensemble [Beel et al., 2016]. Enfin, les jeux de données sont généralement mis à disposition par des producteurs de données bibliographiques ou des fournisseurs de services associés dans un but de recherche. Dans le cadre de la recommandation, ces jeux de données sont surtout utilisés pour l'évaluations hors-ligne des méthodes et des systèmes de recommandation.

Le type de données le plus fréquemment utilisé dans les méthodes analysées est le corpus d'articles scientifiques. Les méta-données disponibles peuvent varier mais on trouve généralement les descriptions bibliographiques, les abstracts, et de manière moins fréquente les citations et les textes complets. Contrairement à ce que l'on pourrait croire, accéder aux citations n'est pas toujours possible et les extraire est un processus complexe et sujet aux erreurs. Ce dernier écueil est cependant à nuancer étant donné l'amélioration des techniques d'extraction et l'augmentation du nombre d'articles disponibles [Lo et al., 2019]. Les principaux jeux de ce type en sciences informatiques sont : *ACL Anthology Network* (AAN) [Radev et al., 2013] [8], *DBLP Computer Science Bibliography* [Ley, 2009] [9], *Citation Network Dataset* (AMINER) [Tang et al., 2008] [10], et *CiteSeerX* [Wu et al.,

8. `http://clair.eecs.umich.edu/aan/index.php`
9. `https://dblp.uni-trier.de/`
10. `https://www.aminer.org/citation`

2015] [11]

D'autres types de données produites par des services académiques sont également disponibles [Beel et al., 2016]. Un premier exemple est le jeu de données *CiteULike* [Wang et al., 2013] extrait de la plateforme éponyme. Il s'agit d'un réseau social académique permettant notamment à ses utilisateurs de partager des articles. Un autre exemple est le jeu de données *Related-Article Recommendation Dataset* (RARD) [Beel, Carevic, Schaible and Neusch, 2017, Beel, Smyth and Collins, 2019] extrait du service de recommandation de littérature scientifique *Mr. DLib* [12]. L'intérêt principal de ces jeux est leur plus grande proximité avec les cas d'utilisation ciblés en recommandation.

Face à ces différents types de jeux de données, le choix doit surtout être motivé par la proximité avec le contexte de production, et notamment les données effectivement utilisées à ce moment-là [Gunawardana and Shani, 2015]. En effet, un phénomène non négligeable est la variabilité des résultats en fonction du jeu de données utilisé [Beel et al., 2016]. De plus, il est important d'être conscient des biais potentiels des différents jeux de données [Gunawardana and Shani, 2015]. Par ailleurs, l'utilisation de plusieurs jeux de données est essentielle pour obtenir des résultats plus robustes [Beel et al., 2016].

3.6 Conclusion

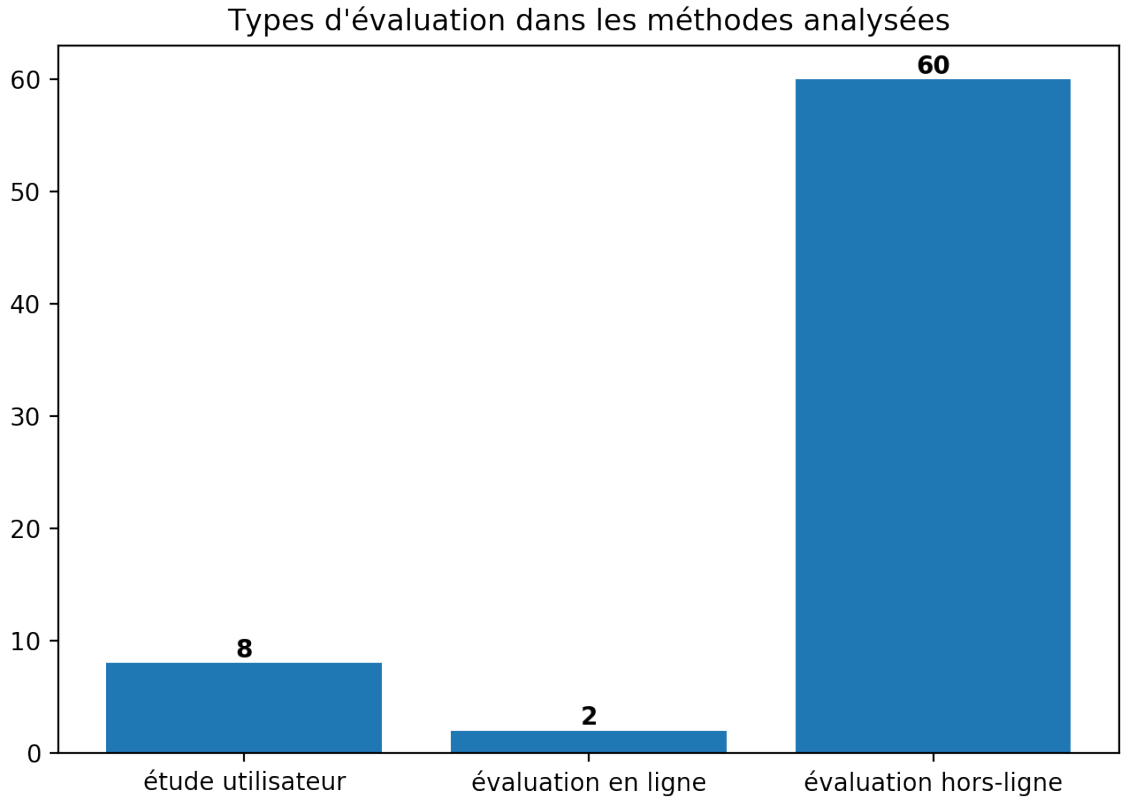

FIGURE 3.1 – Modes d'évaluation utilisés dans les méthodes analysées (voir annexe A.3).

Comme le montre la figure 3.1, l'évaluation hors-ligne est clairement le mode d'évaluation le

11. https://www.cs.uic.edu/~cornelia/citeseer
12. http://mr-dlib.org/

plus populaire dans la littérature. Les raisons principales sont sans doute sa facilité de mise en œuvre et le fait qu'il s'agisse du mode d'évaluation le mieux adapté en début de conception d'une méthode ou d'un système de recommandation. Néanmoins, il convient de ne pas négliger les autres modes d'évaluation dans la suite du développement, notamment parce que de bons résultats lors de l'évaluation hors-ligne ne sont pas une garantie d'efficacité en production [Beel and Langer, 2015]. L'étude utilisateur, qui permet d'obtenir le feedback le plus riche, semble d'ailleurs être une bonne alternative pour valider des résultats comme le montre la figure 3.1.

Enfin, les nombreuses critiques soulevées à l'encontre de l'évaluation hors-ligne, notamment en ce qui concerne les problèmes de réplicabilité, de reproductibilité et la difficulté de généralisation [Beel et al., 2016], mettent en lumière l'absence d'un protocole de référence permettant de comparer les résultats de différentes recherches. Les seules initiatives dans ce sens le jeu de données RARD (extrait d'un véritable système de recommandation) et le framework d'évaluation CITREC. Elles sont cependant assez récentes et donc peut répandues dans la littérature. De plus, il ne s'agit pas encore de protocoles bien définis et dont l'intérêt dans le domaine est démontré.

Chapitre 4

Analyse du système à concevoir

Ce chapitre a pour but de décrire les besoins fonctionnels et non-fonctionnels du système de recommandation à concevoir. La première partie est consacrée à la description du travail d'analyse réalisé. Elle est principalement constituée d'une synthèse du feedback recueilli lors des différentes activités organisées avec des utilisateurs potentiels du système. Elle contient également une partie sur les données potentiellement utilisables.

La seconde partie définit les spécifications proprement dites. Elle débute par la présentation des objectifs de haut niveau du système à réaliser. Ces objectifs sont ensuite déclinés selon différents thèmes qui sont des ensembles de fonctionnalités cohérents. Enfin, un *product backlog* structure le travail d'implémentation proprement dit en déclinant les thèmes en *epics* et *user stories*.

4.1 Synthèse des besoins utilisateurs

4.1.1 Utilisateurs cibles et techniques de collecte utilisées

CETIC

Le CETIC [1] ou Centre d'Excellence en Technologies de l'Information et de la Communication est un centre de recherche appliquée en sciences informatiques. Il a été créé en 2001 par l'UNamur, l'UCLouvain et l'UMons avec pour mission de transposer la recherche en technologiques de l'information et de la communication dans l'industrie wallonne. Il est d'ailleurs reconnu comme Centre de recherche agréé par la Wallonie. Au-delà de ses partenariats industriels, le CETIC participe également à de nombreux projets aux niveaux régional, national et européen.

Le recueil des besoins pour le système de recommandation a été mis en œuvre et piloté par le département COMET. Il a pris la forme d'un parcours utilisateur sur base d'un contexte de recherche librement choisi par chaque participant. Le principe était de décrire les besoins bibliographiques, les outils utilisés et les difficultés rencontrées à chaque étape du contexte choisi. Six personnes ont participé à l'atelier (supervisé par Laurie CECCOTTI [2]) :

— Julien ALBERT, stagiaire,
— Maher BADRI [3], chef de projet, opérations,
— Mathieu GOEMINNE [4], ingénieur de recherche expert, sciences des données,

1. `https://www.cetic.be/`
2. `https://www.cetic.be/Laurie-Ceccotti`
3. `https://www.cetic.be/Maher-Badri`
4. `https://www.cetic.be/Mathieu-Goeminne`

— Simon JACQUET, stagiaire,
— Bérengère NIHOUL [5], ingénieur de recherche, méthodes et outils pour la co-innovation
— et Faiez ZALILA [6], ingénieur de recherche senior, ingénierie logicielle basée sur les modèles et systèmes informatiques distribués.

UNamur - Faculté d'informatique

La Faculté d'informatique de l'UNamur [7] est fondée en 1970. Ses activités principales sont bien sûr l'enseignement et la recherche en sciences informatiques. Elle collabore également avec de nombreux partenaires régionaux, nationaux et internationaux.

Le recueil des besoins a pris la forme d'une interview (voir annexe A.2. Quatre chercheurs de la Faculté ont été interrogés :

— Antoine CLARINVAL [8], doctorant dans le domaine des villes intelligentes,
— Bruno DUMAS [9], professeur d'informatique, spécialisé en interaction homme-machine,
— Nathalie GRANDJEAN [10], chercheuse senior en Science & Technology Studies au Centre de Recherche en Information, Droit et Société (CRIDS),
— et Pierre-Yves SCHOBBENS [11], professeur d'informatique, spécialisé en vérification de logiciel.

4.1.2 Contextes d'utilisation

La recherche bibliographique fait bien sûr partie intégrante du processus de recherche, tant au CETIC qu'à la Faculté d'informatique de l'UNamur. Elle est une étape indispensable à la rédaction d'un projet de recherche, d'un article scientifique, d'une thèse, etc. Elle est également nécessaire dans le cadre de nombreuses autres tâches comme la préparation d'un cours ou d'une formation, ou encore la rédaction d'articles de vulgarisation scientifique. Enfin, la veille est également l'occasion de nombreuses recherches bibliographiques.

Bien que ces nombreux cas d'utilisation possèdent des éléments similaires, leurs spécificités rendent difficile le fait de les prendre tous en compte. Un choix plus raisonnable est de limiter les contextes pris en compte à deux : la rédaction d'un article de recherche et le montage d'un projet de recherche. Il est motivé parce que ces contextes sont plutôt représentatifs des besoins et problèmes de la recherche bibliographique, et qu'ils sont fréquents au sein du CETIC et de l'UNamur. Enfin, la veille bibliographique est également considérée (dans une moindre mesure) car elle permet de mettre en évidence d'autres manières d'effectuer des recherches bibliographiques.

Article scientifique

Dans le cadre de la rédaction d'un article scientifique, l'objectif de la recherche bibliographique est d'identifier les articles clés, les auteurs de référence et les tendances récentes. Il faut pouvoir aller à l'essentiel et montrer la maîtrise du sujet (afin notamment de passer sans encombre l'étape de relecture). Cependant, le sujet d'intérêt est beaucoup plus délimité que dans le cas d'une thèse par exemple et la recherche bibliographique est donc beaucoup plus ciblée. Dans certains cas cependant,

5. https://www.cetic.be/Berengere-Nihoul
6. https://www.cetic.be/Faiez-Zalila
7. https://www.unamur.be/info
8. https://directory.unamur.be/staff/clarinva
9. https://directory.unamur.be/staff/bdumas
10. https://directory.unamur.be/staff/ngrandje
11. https://directory.unamur.be/staff/pyschobb

la recherche bibliographique doit être plus exhaustive, jusqu'à nécessiter la réalisation d'une revue systématique de la littérature. Enfin, un biais en fonction de la question de recherche est souvent introduit afin d'obtenir des références davantage pertinentes.

Projet de recherche

Le projet de recherche consiste généralement à répondre à un appel sur une thématique spécifiée par un organisme clairement identifié. Concrètement, il s'agit de concevoir une proposition de recherche susceptible d'être sélectionnée et permettant donc d'obtenir le financement associé. À première vue, la recherche bibliographique dans le cadre d'un projet de recherche est relativement similaire à un article scientifique. Il existe cependant plusieurs spécificités qui impliquent des besoins distincts.

Tout d'abord, dans le cas d'un projet pour un client bien identifié (ce qui est le plus fréquent au CETIC), il existe plusieurs aspects spécifiques. Premièrement, il faut pouvoir réaliser une mise en contexte en recensant des applications globales et locales (i.e. géographiquement proches) par rapport au domaine d'application du client. La notion de proximité géographique est particulièrement importante pour le CETIC, notamment pour identifier les expériences et les projets locaux, ainsi que les potentiels clients et partenaires.

Ensuite, il est important d'identifier les challenges et les opportunités pour créer de la valeur. Ce qui implique l'exploration de sujets techniquement pointus et, par définition, peu ou pas couverts dans la littérature, et d'idées qui n'ont pas encore été concrétisées. Néanmoins, il existe toujours un risque que l'idée ne soit pas innovante car déjà réalisée ailleurs.

Dans certains cas, il est intéressant de ne pas se limiter aux publications scientifiques mais de pouvoir élargir à d'autres sources plus variées comme de la documentation technique, du code source, des forums, etc. Dans le cas du CETIC, les types de publications envisagés sont sans doute plus variés, avec des méta-données beaucoup moins riches (quand il y en a) et beaucoup plus difficiles à localiser, ce qui pose davantage de problèmes concernant la collecte de l'information.

Veille bibliographique

Concernant la veille bibliographique, les personnes interrogées ont évoqués plusieurs manières de procéder. Tout d'abord, la lecture des conférences et des journaux spécialisés dans le ou les domaines d'intérêt, ainsi que le suivi des publications d'auteurs faisant autorité sont évidemment des canaux importants. Il existe de nombreux outils informatiques disponibles pour la veille bibliographique. Ont été pointés notamment l'utilisation de réseaux sociaux académiques comme *Academia* ou encore *Google Scholar* et ses recommandations générées à partir du profil utilisateur. Enfin, la participation à des conférences et autres événements internationaux permet de se tenir informé. Dans le même ordre d'idée, les différentes discussions avec des collègues, son équipe, ou plus généralement d'autres chercheurs sont également un canal important. Enfin, les recherches ponctuelles réalisées pour des besoins précis permettent indirectement de s'informer sur les dernières tendances.

4.1.3 Quelques aspects importants de la recherche bibliographique

Généralités

Le temps disponible pour réaliser la recherche bibliographique est la contrainte la plus fréquente. Elle est notamment liée aux différentes échéances inhérentes à certains contextes (publication d'un

article, participation à une conférence ou dépôt d'un projet de recherche). De manière générale, les publications les plus récentes sont souvent recherchées lorsqu'il y a une courte échéance de travail et lorsque le domaine concerné évolue très vite. Le besoin principal est donc de rapidement trouver les derniers challenges, les dernières innovations, les problèmes connus, etc. Par contre, les publications plus anciennes sont davantage recherchées lorsque le travail s'inscrit dans une plus longue échéance et qu'il faut découvrir la communauté, l'histoire du domaine, etc.

Une pratique fréquemment relevée indépendamment du contexte est la recherche exploratoire qui permet de voir ce qui existe, d'identifier les éventuelles directions possibles, et de cartographier des connaissances afin d'en avoir une vue d'ensemble. Concrètement, il s'agit d'identifier les concepts importants et la terminologie utilisée (avec ses variations éventuelles), d'identifier les articles de référence et les auteurs clés, de se faire une idée des axes et des potentialités de recherche, et d'identifier la richesse du domaine. Enfin, il est important de noter que l'évaluation de la pertinence des références est souvent plus difficile dans ce cas. Un aspect lié est la capacité à retrouver les différents points de vue associés à un domaine de recherche, avec notamment le risque d'être parfois hors sujet.

Identification des articles, auteurs, journaux et conférences de référence

Un challenge très fréquemment soulevé est l'identification des articles, auteurs, journaux et conférences de référence. En général, les personnes interrogées cherchent plutôt à identifier les auteurs de référence. Les conférences ou les journaux sont moins essentiels, bien que cela puisse varier en fonction du domaine d'intérêt. À noter cependant l'importance de certaines maisons d'édition ou collections prestigieuses qui font malgré tout autorité dans certains domaines, notamment en sciences humaines. Les auteurs sont souvent représentatifs de certains positionnements dans la recherche, voire de certaines écoles de pensée. De plus, cette approche favorise l'interdisciplinarité.

Dans le cas d'un auteur, d'une conférence ou d'un journal, un intérêt immédiat est la consultation de la bibliographie associée. Dans le cas d'un article clé, outre sa bibliographie, ses citations peuvent également être intéressantes.

Recherche par mots-clés

Un autre aspect important est la recherche par mots-clés. La plupart du temps, celle-ci est employée pour amorcer une recherche bibliographique. Elle est aussi utilisée pour effectuer une recherche très précise ou pour localiser un article spécifique. Enfin, elle est bien sûr incontournable dans le cas d'une revue systématique de littérature.

La capacité à trouver les bons mots-clés est néanmoins une difficulté fréquemment soulevée. Une première explication est due aux problèmes classiques liés à l'utilisation de mots-clés comme la polysémie, l'homonymie ou les variations lexicales ou grammaticales. Plus spécifiquement dans le cadre de la recherche, il existe un phénomène de variation terminologique qui implique que plusieurs termes, ou combinaisons de termes, peuvent correspondre à un même sujet de recherche. Les deux causes principales sont le nombre parfois important de contributeurs à un même domaine et l'évolution générale de la recherche au fil du temps. Enfin, un phénomène plus récent est l'utilisation de *buzzwords* en provenance de l'informatique et plus spécifiquement de l'intelligence artificielle qui accentue cette difficulté.

Saturation/Maturité

Parmi les autres aspects évoqués, il y a la notion de saturation dans un domaine de recherche, c'est-à-dire le fait qu'un domaine ait déjà fait l'objet de nombreuses publications et qu'une nouvelle recherche ne semble pas avoir beaucoup de sens. Elle peut aussi indiquer une certaine maturité de celui-ci et donc un élément intéressant pour évaluer le potentiel de mise en application dans l'industrie.

4.1.4 Expériences avec un système de recommandation

De manière générale, les expériences ne sont pas très positives, notamment parce que les recommandations intéressantes sont très rares. Le manque de précision des recommandations est d'ailleurs souvent pointé comme la raison d'abandon. De plus, le contexte dans lequel celles-ci sont proposées n'est pas toujours opportun, la réception périodique de mails est un exemple cité (par ex. *Mendeley*[12]).

Les recommandations d'outils comme *Academia*[13], *ResearchGate*[14] ou *Mendeley* ne sont en général pas très appréciées car souvent connues ou hors sujet. Certains participants pointent toutefois l'intérêt d'*Academia* pour sa dimension réseau social, permettant notamment de suivre d'autres chercheurs et de voir les articles avec lesquels ils interagissent dans le flux d'actualité, ce qui favorise notamment la sérendipité. *Arxiv-Sanity*[15], qui était le point de départ de ce travail, n'est pas très connu et n'a guère convaincu ceux qui l'ont utilisé.

Google Scholar[16] semble toutefois sortir du lot. Son système de recommandation via le profil de l'utilisateur fonctionne assez bien selon les participants qui le connaissent et l'utilisent. Il est tout de même important de maintenir son profil à jour et de le nettoyer afin de maintenir la pertinence des recommandations, ce qui peut être fastidieux.

Enfin,les systèmes de recommandation ne sont pas vraiment privilégiés dans un processus de recherche bibliographique selon les participants. En effet, la recherche bibliographique est plutôt perçue comme une démarche proactive pour laquelle ce mode d'interaction ne convient pas. Enfin, l'explicabilité du système de recommandation est un élément important pointé par plusieurs personnes mais guère présent dans les systèmes existants.

4.1.5 Fonctionnalités souhaitées

Idéalement, l'outil à développer doit être intégré au processus de recherche et pas un outil supplémentaire à part. De plus, il ne doit pas chercher à remplacer un outil donnant déjà pleine satisfaction à ses utilisateurs (par exemple *Google Scholar*).

La gestion des résultats est également un élément important. Des outils d'aide pour la gestion et l'analyse d'un corpus de références sont souhaités par les participants. Une idée évoquée est d'aider à la création et à l'analyse de tableaux de classification des résultats, par exemple en identifiant des axes très développés (indication de maturité), ou au contraire des pistes peu explorées (recherche d'innovation).

De manière générale, La lisibilité de l'interface est également un élément primordial (*Google Scholar* est d'ailleurs fréquemment cité en exemple). Concernant l'affichage des résultats proprement

12. `https://www.mendeley.com/`
13. `https://www.academia.edu/`
14. `https://www.researchgate.net/`
15. `http://www.arxiv-sanity.com/`
16. `https://scholar.google.com/`

dit, d'une part, le fait de ne pas montrer trop de résultats semblent faire consensus. Il faut également pouvoir les trier et les filtrer selon différents critères et modalités (la pertinence et la date de publication sont les plus fréquemment cités).

Littérature sous forme de graphe

La représentation de la littérature sous forme de graphe des citations est souvent évoquée dans les discussions. La motivation principale serait de pouvoir explorer visuellement un corpus d'articles afin d'identifier les nœuds et les relations importants, voire les éventuels *clusters*. Un outil de recherche sous forme de graphe a été également évoqué mais les réactions sont plus mitigées. Au-delà de la visualisation, la représentation de la littérature sous forme de graphe permettrait également l'utilisation de mesures et techniques spécifiques (centralité, PageRank, etc.) qui pourraient s'avérer intéressantes pour l'analyse du domaine.

La modélisation sous forme de graphe ne doit pas nécessairement se limiter aux citations. En effet, elle pourrait être élargie en variant les types de nœuds (auteurs, journaux, conférences, mots-clés, topics, etc.) et les types de relations (citations, indexation, contributions, etc.). Les possibilités d'analyse seraient par conséquence beaucoup plus larges et permettrait de comprendre en profondeur les domaines d'intérêt. Enfin, l'utilisation des graphes pour la gestion et l'analyse des références obtenues via une revue systématique de littérature est également considérée comme intéressante.

4.2 Analyse orientée données

Outre l'analyse classique orientée utilisateur, il est également important de s'intéresser aux données qui vont être manipulées par l'application. Il s'agit d'analyser la sémantique, la volumétrie, les caractéristiques, l'accessibilité, etc. Cette analyse a pour but *in fine* de fixer une hypothèse de travail quant aux données manipulées et de poser des choix de conception éclairés, notamment en ce qui concerne l'architecture et les méthodes de recommandation employées. D'une certaine manière, le développement de l'application doit être piloté également par les données.

Définition du corpus

Le corpus des items ciblé par le système de recommandation est l'ensemble des publications scientifiques en sciences informatiques. Afin d'éviter les écueils liés à la définition de publication scientifique ou à la délimitation des sciences informatiques, le plus simple est de délimité le corpus de manière opérationnelle. Sont donc prises en compte les références bibliographiques produites par les principales bases de données bibliographiques existantes et pouvant être catégorisées comme appartenant au domaine des sciences informatiques.

Volumétrie

Il est difficile de connaître précisément le nombre total de publications scientifiques. Néanmoins, il existe différentes estimations du nombre de références présentes sur les moteurs de recherche spécialisés qui permettent de s'en faire une idée. Par exemple, Gusenbauer [2019] estime que *Google Scholar* contient 389 millions de références. En considérant l'hypothèse que *Google Scholar* échoue à capturer 13 % des publications en ligne [Khabsa and Giles, 2014], cela donne le chiffre total de 447 millions de références.

Microsoft Academic[17] annonce 240 millions de références indexées et 23 millions pour les références en sciences informatiques (*Computer science*). Ce qui donne une ratio de 9,6 % et donc une estimation de près de 43 millions pour l'ensemble des références existantes.

Enfin, le nombre de publications scientifiques est également en constante augmentation. Le taux d'accroissement annuel est généralement estimé à environ 3 % [Jinha, 2010]. Cependant, Bornmann and Mutz [2015] évoque un taux d'accroissement annuel de 8 à 9 % à partir de la seconde guerre mondiale. Il expliquent notamment cette accélération par l'importance du développement de la recherche scientifique en Asie.

Accessibilité des articles

De manière générale, les contraintes d'accès aux publications scientifiques et le coût pour les acquérir sont problématiques. La question de la présence de références à des publications non-accessibles est donc légitime. Néanmoins, il est important de mentionné que d'après Khabsa and Giles [2014], 24 % des publications scientifiques sont librement accessibles en ligne. Et ce chiffre monte à 50 % pour les publications en sciences informatiques. De plus, les bibliothèques universitaire disposent en général d'abonnements permettant d'augmenter le nombre de publications accessibles. Et la possibilité de contacter un auteur pour se procurer un article n'est pas non plus à négliger. Enfin, il est de toute façon important qu'un chercheur puisse avoir une vue sur l'ensemble de la production scientifique.

4.2.1 Sources de données

Outre les jeux de données évoqués lors de l'évaluation hors-ligne (voir section 3.5) qui sont souvent loin d'être complets et font souvent l'objet de conditions d'utilisation limitant leur emploi à un contexte de recherche, il est souvent nécessaire de passer par d'autres sources de données lors de la mise en production du système de recommandation. Les API et autres sources de données sont disponibles pour pouvoir constituer un corpus de publications scientifiques utilisable en production par un système de recommandation. Ces sources se distinguent des jeux de données car le corpus mis à disposition est ouvert (et mis-à-jour régulièrement) et n'est pas téléchargeable d'un seul bloc. Elles ne sont donc pas adaptées à l'évaluation. Il s'agit principalement d'API mises à disposition par des producteurs de données bibliographiques.

Comme le montre Chen [2010], ces sources de méta-données couvrent 95 % des publications scientifiques. Néanmoins, la constitution d'un corpus d'articles de taille suffisante implique d'agréger de multiples sources. Ce qui implique d'une part de pouvoir évaluer la qualité des méta-données des sources utilisées, et d'autre part de mettre en place un processus d'intégration et de consolidation gérant les problèmes typiques de ce genre de processus (dédoublonnage, gestion des données contradictoires, données manquantes ou erronées, etc.). Enfin, certaines méta-données comme les citations ne sont pas toujours disponibles.

Liste des principales sources de données identifiées (Attention aux conditions d'utilisation !) :

— Arxiv API, `https://arxiv.org/help/api`
— Semantic Scholar API, `http://api.semanticscholar.org/`
— Researchr API, `https://researchr.org/about/api`

17. `https://academic.microsoft.com/`, consulté le 25/07/2020

— IEEE Xplore API Portal, https://developer.ieee.org/
— CiteSeerX, https://csxstatic.ist.psu.edu/downloads/data
— Mendeley, https://dev.mendeley.com/
— Core, https://core.ac.uk/services/#access-to-raw-data
— Dimensions Runtime API, https://www.dimensions.ai/dimensions-apis/
— Web of Science Group APIs, https://clarivate.com/webofsciencegroup/solutions/xml-and-apis/
— CrossRef REST API, https://www.crossref.org/services/metadata-delivery/rest-api/
— Microsoft Academic API, https://academic.microsoft.com/home

4.3 Spécifications

4.3.1 Application

La proposition initiale du CETIC consistait en la réalisation d'un système capable de recommander à un utilisateur un ensemble de publications pertinentes sur base d'une sélection de références bibliographiques fournies par ce dernier. Il s'agissait de concevoir un système de recommandation en temps réel prenant la forme d'une application Web. Le principe de recommandation envisagé était de type « one-shot », c'est-à-dire que l'utilisateur soumet une sélection d'articles en entrée et reçoit une série de recommandations en retour (à l'image d'*Arxiv-Sanity*, http://www.arxiv-sanity.com/). Au fil des discussions, ce principe est cependant remis en cause. D'une part, à cause du manque d'intérêt manifesté par les personnes interrogées pour cette formule, et d'autre part parce que les objectifs et pratiques de recherche bibliographique identifiés semblent montrer qu'un système davantage interactif est plus adéquat.

Un scénario d'interaction de type « ping-pong » entre l'utilisateur et le système de recommandation a donc été privilégié. L'idée est de proposer à l'utilisateur une première sélection large de références bibliographiques et, sur base de son feedback (par exemple sous forme d'une sélection de références qui l'intéressent), de lui proposer de nouvelles recommandations, et ainsi de suite... Le système déduit donc les thématiques d'intérêt à partir des interactions de l'utilisateur. Celui-ci peut également suggérer de nouvelles directions au fur et à mesure des itérations et privilégier d'autres objectifs de recherche bibliographique. Enfin, il peut également manipuler la liste des résultats en fonction de ses objectifs (via des filtres et des tris).

Objectifs de l'application

Le système doit donc aider l'utilisateur à répondre à ses besoins bibliographiques. Les objectifs identifiés lors de l'analyse sont les suivants :

1. **Compréhension générale du domaine** : Le système permet d'avoir une vue d'ensemble du domaine, de cartographier des connaissances, et d'identifier les concepts principaux, les différents scopes et enjeux, etc. par le biais des recommandations proposées.
2. **Identification des ressources de référence** : Le système met en évidence les articles de référence, mais également les auteurs, les journaux et les conférences importants.
3. **Découverte des challenges et des opportunités** : Le système propose également des recommandations relatives aux dernières innovations, tendances, idées à concrétiser, challenges

et opportunités. Les articles recommandés ici sont davantage récents, novateurs, davantage exploratoires, et idéalement inconnus pour l'utilisateur.

4. **Localisation de la recherche** : Le système est capable de situer géographique la production scientifique.
5. **Identification de la terminologie** : Le système présente les mots-clés du domaines et leurs variations
6. **Évaluation de la richesse du domaine** : Le système donne un aperçu à l'utilisateur de la richesse du domaine et de sa maturité/saturation.
7. **Recommandation en temps-réel** : Sur le plan non-fonctionnel, le système est capable de proposer des recommandations en temps réel à l'utilisateur.

4.3.2 Thèmes

L'objectif de cette section est de découper l'application en groupes cohérents d'éléments fonctionnels et non-fonctionnels appelés thèmes. Il y en a quatre :

— la génération des recommandations,
— l'exploitation des recommandations,
— la constitution du corpus des articles,
— l'intégration dans le processus de recherche.

Génération des recommandations

La génération des recommandation comprend les différents aspects fonctionnels et non-fonctionnels afférents. Tout d'abord, le système est capable de générer des recommandations en temps réel à partir d'un profil utilisateur caractérisé notamment par des références bibliographiques. L'utilisateur obtient rapidement des résultats pertinents selon ses différents objectifs de recherche bibliographique, notamment via différentes possibilités de paramétrage. Il est capable d'explorer la littérature de manière ciblée en orientant les recommandations au fil des itérations. Mais même si les objectifs peuvent être variés, la précision (entendue comme la similarité sémantique avec le sujet les articles en entrée) des recommandations reste un critère primordial. Enfin, la lisibilité de l'interface, particulièrement en ce qui concerne la présentation des résultats, et l'explicabilité des recommandations sont également importantes.

Exploitation des résultats

L'exploitation des résultats comprend la présentation, la gestion et l'exploitation des articles proposés. Le système favorise ici aussi l'obtention rapide de résultats pertinents selon les différents objectifs de l'utilisateur grâce à différentes possibilités de filtrage et de tri. Au-delà des articles, l'utilisateur peut également identifier d'autres types d'entités importantes comme des auteurs, des mots-clés, des journaux ou des conférences. Il peut également exploiter les possibilités offertes par une modélisation des données sous forme de graphe (visualisation des résultats, techniques spécifiques commme des mesures de centralités, etc.). Et il peut sauvegarder des articles sous forme d'un panier de références. L'explicabilité des fonctionnalités et la lisibilité de l'interface reste particulièrement importantes ici aussi.

Constitution d'un corpus de références

La constitution d'un corpus des références bibliographiques consiste en l'intégration des publications scientifiques en informatique (quelques dizaines de millions de références) dans un modèle de données permettant de générer des recommandations. Dans un premier temps, le corpus est limité aux articles scientifiques mais pourra être étendu par la suite à d'autres documents techniques et/ou issus de la recherche, du code source, etc. Les aspects importants du corpus sont la qualité des méta-données, la prise en compte de multiples sources de données et la présentation sous les formes les plus pertinentes (notamment sous forme de graphe). Enfin, l'incidence du volume des données sur les besoins non-fonctionnels de l'application, notamment la recommandation temps réel, doit être gérée.

Intégration dans le processus de recherche

Le quatrième et dernier thème est l'intégration du système dans le processus de recherche, c'est-à-dire avec les autres outils utilisés. L'utilisateur peut importer une bibliographie au format BibTeX comme entrée pour le système de recommandation et exporter les résultats obtenus au format BibTeX. Les possibilités d'entrées pourront s'étendre à une ébauche d'article au format latex voire un pdf. Le système doit également dialoguer avec d'autres outils utilisés pour la recherche bibliographique, comme *Google Scholar*, ou pour d'autres tâches comme la rédaction, avec *Overleaf*[18] par exemple, afin d'extraire les informations pertinentes pour la construction du profil de l'utilisateur et d'exporter les résultats.

4.3.3 *Product backlog*

Afin de structurer l'implémentation du prototype, l'application est découpée en *epics* et en *user stories*. Les *user stories* sont des fonctionnalités vues depuis l'utilisateur et de taille suffisamment réduite afin de permettre une implémentation rapide. Les *epics* sont des *user stories* de plus haut niveau, c'est-à-dire qu'elles décrivent des fonctionnalités plus étendues, et elles ne peuvent pas être implémentées rapidement. Enfin, la notion de MVP (et MVP+1, MVP+2, etc.) permet de distinguer différentes versions du système et d'organiser son implémentation en sprints de la manière suivante :

- **MVP** : mise en place et premiers essais avec un jeu de données de petite taille (AAN).
- **MVP+1** : passage à la recommandation sur base de plusieurs articles, mise en place du pipeline définitif (recherche → recommandation → résultats), mise en place de l'architecture définitive.
- **MVP+2** : mise en place de la stratégie d'optimisation selon les objectifs de l'utilisateur.
- **MVP+3** : prototype final.
- **MVP+99** : fonctionnalités non prioritaires pour le prototype.

Les différentes *stories* sont visibles dans les tables 4.1, 4.2, 4.3 et 4.4. Ces tables font office de journal de développement du prototype. Certaines fonctionnalités ne sont donc plus accessibles dans la version finale.

18. `https://www.overleaf.com/`

TABLE 4.1 – Génération des recommandations

Epic	US	MVP	Description	Fait
1	Construction du profil de l'utilisateur lors de la première itération (références + paramétrisation)			
1	1	MVP	Encodage d'une référence	√
1	2	MVP+1	Encodage de plusieurs références	√
1	3	MVP+2	Paramétrisation	
2	Construction du profil de l'utilisateur lors des itérations suivantes (références + paramétrisation)			
2	1	MVP	Sélection d'une référence	√
2	2	MVP+1	Sélection de plusieurs références	√
2	3	MVP+2	Paramétrisation	
3	Génération des recommandations			
3	1	MVP	Génération des recommandations à partir d'une référence (TF-IDF + similarité cosinus)	√
3	2	MVP+1	Génération des recommandations à partir de plusieurs références (Personalized PageRank)	√
3	3	MVP+1	Génération des recommandations à partir de plusieurs références (BM25, Elasticsearch)	√
3	4	MVP+1	Génération des recommandations à partir de plusieurs références (citations (Jaccard), PostgreSQL)	√
3	5	MVP+1	Génération des recommandations à partir de plusieurs références (Personalized PageRank, Neo4J)	√
3	6	MVP+1	Génération des recommandations à partir de plusieurs références (DiSCern)	√
3	7	MVP+2	Génération des recommandations selon des paramètres (TF-IDF, Elasticsearch)	√
3	8	MVP+2	Génération des recommandations selon des paramètres (citations (Jaccard), Elasticsearch)	√
3	9	MVP+2	Génération des recommandations selon des paramètres (mots-clés (Jaccard), Elasticsearch)	√
3	10	MVP+2	Génération des recommandations selon des paramètres (stratégie multi-objectifs)	
3	11	MVP+2	Génération des recommandations selon des paramètres	
4	Évaluation en ligne (choix aléatoire des méthodes de recommandation + logging)			
4	1	MVP+99	Logging	
4	2	MVP+99	Choix aléatoire de la méthode de recommandation (testing A/B)	
5	Explicabilité des recommandations			

Table 4.2 – Exploitation des recommandations

Epic	US	MVP	Description	Fait
6	Présentation des résultats			
6	1	MVP	Présentation des résultats sous forme de liste	√
6	2	MVP	Affichage des détails d'un référence	√
7	Tris			
7	1	MVP+2	Tri selon différents critères	√
8	Filtres			
8	1	MVP+2	Filtre textuel intégré aux résultats	√
8	2	MVP+2	Filtre selon autres critères	
9	Mise en évidence des entités pertinentes			
9	1	MVP+3	Mise en évidence d'un type d'entités (une story par type)	√
10	Persistance			
10	1	MVP+99	Sauvegarde du profil utilisateur	
10	2	MVP+99	Sauvegarde des résultats	
11	Explicabilité des fonctionnalités			
12	Présentation des résultats (graphe)			
13	Évaluation en ligne (choix aléatoire des méthodes + logging)			

Table 4.3 – Constitution d'un corpus de références

Epic	US	MVP	Description	Fait
14	Constitution d'un corpus à partir du dataset AAN (20k références)			
14	1	MVP	Utilisation du dataset d'évaluation	√
15	Constitution d'un corpus à partir du dataset DBLP-AMiner (4M références)			
15	1	MVP+1	Utilisation du dataset d'évaluation	√
15	2	MVP+2	Utilisation du dataset complet	√
16	Constitution d'un corpus à partir d'une source de référence en lignes (20M références)			
16	1	MVP+3	Constitution d'un corpus de départ	
16	2	MVP+3	Système de mise-à-jour continu du corpus (via une source de références)	
17	Intégration de sources multiples et enrichissement des métadonnées			

Table 4.4 – Intégration dans le processus de recherche (notamment vis-à-vis des autres outils utilisés)

Epic	US	MVP	Description	Fait
18	Accès à la référence			
18	1	MVP+3	Lien vers la ressource (via DOI)	√
19	Exportation des résultats au format BibTeX			
19	1	MVP+3	Exportation des résultats complets au format BibTeX	√
19	2	MVP+3	Exportation d'une sélection de résultats au format BibTeX	√
20	Importation d'une bibliographie BibTeX pour définir le profil de l'utilisateur			
20	1	MVP+3	Importation des résultats complets au format BibTeX	√
20	2	MVP+3	Importation d'une sélection de résultats au format BibTeX	√
21	Importation d'un fichier latex pour définir le profil de l'utilisateur			
22	Importation d'un pdf pour définir le profil de l'utilisateur			
23	Importation d'informations à partir de Google Scholar pour construire le profil utilisateur			
24	Exportation des résultats vers Overleaf			

Chapitre 5

Comparaison et sélection des méthodes

Ce chapitre aborde la comparaison et la sélection des méthodes pour le développement du prototype. Il est constitué de trois parties : la description du protocole d'évaluation, la présentation des méthodes comparées, et l'analyse des résultats. Le code source de l'évaluation est disponible dans l'annexe A.5.

5.1 Protocole d'évaluation hors-ligne

L'objectif du protocole d'évaluation est de pouvoir comparer des méthodes appartenant à différentes classes de recommandation afin de déterminer la où les méthodes les plus à même de répondre aux exigences du système à concevoir. Le mode d'évaluation hors-ligne est choisi pour sa facilité de mise en œuvre et parce qu'il est parfaitement adapté à cet objectif. Concrètement, il s'agit donc de définir les jeux de données utilisés et la manière dont ils sont employés, de présenter les aspects étudiés avec les mesures associées, et enfin de décrire le processus d'analyse des résultats.

5.1.1 Données utilisées

Le premier jeu de données utilisé est l'*ACL Anthology Network* (AAN) [Radev et al., 2013] dans sa version 2014 et publié en décembre 2016 [1]. Il comprend 23 766 articles, 18 862 auteurs, 373 lieux de publication [2] et 124 857 citations. Il est construit à partir des pdf orignaux de l'anthologie de l'Association for Computational Linguistics [3]. Cette anthologie reprend une sélection d'articles relatifs à la linguistique informatique et au traitement automatique du langage naturel.

Le second jeu de données utilisé est le *DBLP-Citation-network* (DBLP) [Tang et al., 2008] dans sa version 12, publié le 9 avril 2020 [4]. Il comprend 4 894 091 articles et 45 564 149 dans sa version complète. Il s'agit d'un ensemble de publications en sciences informatiques extraites principalement depuis DBLP, ACM et MAG (*Microsoft Academic Graph*). Afin de faciliter l'évaluation des recommandations, un sous-ensemble traitant du deep learning a été sélectionné via le mot-clé *deep*

1. `http://clair.eecs.umich.edu/aan/index.php`
2. Au sens de revue ou de conférence (traduit de *venue*)
3. `https://www.aclweb.org/anthology/`
4. `https://www.aminer.org/citation`

learning. Les mots-clés (*fields of study*) proviennent du MAG où les articles sont indexés de manière semi-automatique [Shen et al., 2018]. Chaque mot-clé est également associé à un poids qui correspond au score de similarité entre le mot-clé et l'article.

Le choix de ces jeux de données est principalement motivé par le fait qu'ils sont fréquemment utilisés dans la littérature, et parce que les méta-données disponibles permettent d'évaluer une variété importante de méthodes. De plus, l'utilisation de deux jeux différents est une garantie d'obtenir des résultats plus robustes, en atténuant la variabilité des résultats fréquemment constatée avec l'évaluation hors-ligne [Beel et al., 2016].

Pour chaque jeu, le preprocessing est réalisé en deux étapes. La première consiste en la reconstruction des méta-données (i.e. identifiants, titres, abstracts, auteurs, lieux et années de publication) à partir des fichiers de départ. À noter que l'extraction des abstracts a été particulièrement laborieusement dans le cas d'AAN car il a fallu utiliser les textes complets des articles océrisés. La seconde étape a pour but de sélectionner uniquement les articles ayant des méta-données suffisamment complètes, c'est-à-dire un identifiant, un titre, un abstract, au moins un auteur, une année de publication, un mot-clé et au moins une citation entrante ou sortante. Dans le cas de AAN, les mots-clés sont ajoutés depuis DBLP via correspondance entre les titres des articles. Enfin, seuls les articles appartenant à la plus grande composante connexe du graphe non orienté des citations sont retenus. Au final, il reste 12 274 articles pour AAN et 22 726 articles pour DBLP regroupés dans deux fichiers au format json.

Pour chaque jeu, un sous-ensemble de test de 2000 entrées est constitué de la manière suivante. Tout d'abord, 2000 articles ayant au moins 3 références bibliographiques sont choisis aléatoirement et sans remise. Et pour chaque article, l'entrée est construite à partir de 3 articles choisis aléatoirement et sans remise parmi sa bibliographie. Plusieurs manières de faire étaient possibles pour la constitution du jeu de test, comme choisir simplement un ou plusieurs articles de manière aléatoire. Cependant, le choix de cette méthode est motivé par la volonté d'être proche de cas d'utilisation réels. En effet, considérer plusieurs articles, trois en l'occurrence, en provenance d'une même bibliographie correspond assez bien au cas d'utilisation type qui est d'alimenter une bibliographie. Enfin, ce choix offre des garanties quant au sens à combiner ces articles pour recommandation (contrairement à des articles choisis aléatoirement et n'ayant potentiellement rien en commun).

5.1.2 Aspects et mesures évalués

L'objectif principal est de pouvoir comparer des méthodes de recommandation appartenant à différentes classes. L'utilisation d'une approche supervisée risque donc de ne pas être satisfaisante (voir section 3.4.2). L'approche choisie consiste donc à évaluer de manière non-supervisée la qualité des recommandations selon différents aspects correspondant aux besoins de l'utilisateur. Concrètement, il s'agit de générer des listes de 100 recommandations à partir des entrées des jeux de test et de comparer les résultats obtenus par les différentes méthodes selon les mesures décrites ci-dessous.

Précision

Le premier aspect envisagé est la précision des recommandations, c'est-à-dire la similarité sémantique des articles recommandés avec les articles en entrée. La première mesure utilisée est basée les méta-données d'indexation. Elle consiste en la moyenne des similarités cosinus entre les vecteurs de mots-clés des recommandations et le vecteur des mots-clés obtenu par concaténation des vecteurs des articles en entrée (lorsque un mot-clé est présent plusieurs fois, la moyenne des poids est prise

comme poids final). La similarité cosinus sim_{cos} est calculée selon la formule suivante :

$$sim_{cos}(V_1, V_2) = \frac{V_1 \cdot V_2}{||V_1||||V_1||}$$

V_1 et V_2 étant deux vecteurs de mots-clés. Le choix d'une mesure basée sur les mots-clés est inspiré de l'utilisation du MeSH par le framework CITREC [Gipp and Meuschke, 2015]. Cette mesure permet d'éviter les biais induits par les mesures basées sur la similarité textuelle. Ces dernières risquent en effet de favoriser les méthodes de recommandation basées sur le contenu (voir section 3.4). Cette mesure est évidemment le principal critère de sélection des méthodes étant donné l'importance de ce facteur pour les utilisateurs potentiels (voir chapitre 4).

La seconde mesure de précision est basée sur les citations. Elle consiste en la moyenne des indices de Jaccard entre les ensembles des références et citations des recommandations et l'ensemble des références et citations des articles en entrée. Cette mesure est aussi appelé similarité de Amsler dans ce cas. L'indice de Jaccard sim_{jac} est calculé de la manière suivante :

$$sim_{jac}(RC_1, RC_2) = \frac{|RC_1 \cap RC_2|}{RC_1 \cup RC_2}$$

avec deux ensembles de références et citations RC_1 et RC_2. Cette seconde mesure de précision a surtout pour but de comparer les méthodes exploitant le graphe des citations aux autres.

Diversité

Le deuxième aspect envisagé est la diversité des recommandations, entendue comme la variété des articles d'une liste de recommandations. Elle s'appuie sur l'évaluation de la dissimilarité moyenne entre les différents items d'une liste. La diversité moyenne d'un ensemble de recommandations R est obtenue de la manière suivante :

$$div(R) = \frac{\sum_{i \in R} \sum_{j \in R \setminus \{i\}} dis(i,j)}{|R|(|R|-1)}$$

La dissimilarité $dis(i,j)$ est obtenue à partir du complément de la similarité sim :

$$dis(i,j) = 1 - sim(i,j)$$

Deux mesures sont employées : le complément de la similarité cosinus sur les vecteurs de mots-clés et le complément de l'indice de Jaccard sur les ensembles des références et citations. La première mesure a surtout pour but d'évaluer la capacité des méthodes à identifier les diverses approches par rapport à un sujet d'intérêt. Tandis que la seconde cherche plutôt à évaluer la capacité des méthodes à localiser les différents clusters d'articles sur un même sujet.

Nouveauté

Le troisième aspect envisagé est la nouveauté des recommandations. Cette notion comprend évidemment le caractère récent des articles. La mesure utilisée dans ce cas est la date de publication moyenne des recommandations pour une liste donnée. Son but est surtout d'identifier l'éventuelle tendance à favoriser les articles plus anciens (et donc ayant plus de citations) de certaines méthodes.

La nouveauté peut être également interprétée comme le fait d'être peu ou pas connu de la communauté des chercheurs. Dans ce cas, la mesure utilisée est la moyenne de l'inverse de la

popularité, approximée par le nombre de citations. Elle est calculée de la manière suivante :

$$nouv(R) = \frac{\sum_{i \in R} -log_2 p(i)}{|R|}$$

avec R une liste d'items recommandés et $p(i)$ le nombre de citations de l'item i.

Couverture

Enfin, le dernier aspect envisagé est la couverture globale des méthodes de recommandation. Celle-ci est envisagée du point de vue des utilisateurs et est alors définie comme le pourcentage d'utilisateurs, représentés par les différents entrées du jeu de test, pour lesquelles une méthode renvoie une liste complète de recommandations (i.e. contenant 100 items). Le but est d'identifier les méthodes incapables de proposer des recommandations à un nombre suffisant d'utilisateurs.

La couverture globale est également envisagée du point de vue des articles et est alors définie comme le rapport entre la taille de l'ensemble des articles recommandés pour les différentes entrées et la taille du corpus entier. Le but est d'identifier les méthodes ayant tendance à recommander les mêmes items

5.1.3 Analyse des résultats

Comme le rappelle notamment Gunawardana and Shani [2015], il est important de réaliser des tests statistiques pour déterminer si les différences de performance constatées sont bien significatives. Il convient tout d'abord de voir si l'hypothèse nulle (i.e. les performances des différentes méthodes sont similaires) peut être rejetée et donc que les différences de performance de certaines méthodes sont significatives. Les tests réalisés sont choisis sur base de Demšar [2006] pour le cas où plusieurs méthodes sont comparées.

Tout d'abord, un test de Friedman est réalisé. Dans le cas où la *p-value* obtenue est inférieure à 0,05, un test post-hoc peut être réalisé afin de distinguer les performances des différentes méthodes. Le test choisi est le test de Nemenyi. Son principe est de considérer que deux méthodes ont des performances significativement différentes si leurs rangs moyens sont différents d'une valeur minimum appelée *différence critique*.

5.2 Méthodes candidates

5.2.1 Méthodes de base

Les méthodes de base choisies sont représentatives des différentes classes employées en recommandation de littérature scientifique et sont fréquemment utilisées comme références dans la littérature. L'idée est de pouvoir les comparer avec des méthodes *state-of-art* afin de déterminer si la plus grande complexité de ces dernières se justifie en terme de performances.

TF-IDF + similarité cosinus

Cette méthode est basée sur le contenu et utilise la technique de représentation vectorielle TF-IDF [Jones, 1972]. L'implémentation utilisée est celle de *Scikit-learn*[5]. Le principe est de construire

5. `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html`

un vecteur pour chaque article (à partir du titre et de l'abstract dans le cas présent) à partir des fréquences d'apparition des mots dans cette article, pondérées par la fréquence d'apparition inverse de ces mots dans les articles du corpus. La valeur du TF-IDF [6] pour un terme t et un document d est exprimée de la manière suivante :

$$\textit{tf-idf}(t,d) = \textit{tf}(t,d) \times \textit{idf}(t)$$

avec comme fréquence du terme dans le document :

$$\textit{tf}(t,d) = \frac{\text{nombre d'occurences de } t \text{ dans } d}{\text{nombre de mots dans } d}$$

ou, et comme fréquence inverse du mot dans le corpus :

$$\textit{idf}(t) = log\frac{1+n}{1+\textit{df}(t)} + 1$$

où n est le nombre de documents dans le corpus et $\textit{df}(t)$ est le nombre de documents contenant le terme t. L'idée est de prendre en compte respectivement l'importance de ce mot dans le document et sa rareté dans le corpus. Une fois les vecteurs obtenus, la similarité cosinus est utilisée pour comparer les articles. Deux variantes sont utilisées : la première fait la somme des scores de similarité avec les articles en entrée pour chaque article candidat, et la seconde construit un vecteur à partir de la concaténation des titres et des abstracts des articles en entrée.

BM25

Cette méthode est également basée sur le contenu et utilise la technique de représentation vectorielle BM25 [Robertson and Zaragoza, 2010]. Il s'agit d'une méthode similaire à TF-IDF utilisant la fréquence d'apparition des mots dans les documents pour pouvoir les comparer. Elle est ici appliquée aux titres et abstracts des articles. L'implémentation de la librairie *Gensim* [7] est utilisée et la fonction de scoring [8] y est définie comme :

$$score(D,Q) = \sum_{i=1}^{n} \frac{freq(q_i,D)\cdot(k_1+1)}{freq(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)} \cdot idf(q_i,\mathcal{D})$$

Elle prend comme paramètres un document D appartenant au corpus $\mathcal{D}$ et une requête Q composée de n mots q_i (et qui est ici un autre document du corpus $\mathcal{D}$). $freq(q_i,D)$ est la fréquence du mot q_i dans le document D, et k_1 et b sont des paramètres fixés ici à 1.5 et 0.75 , $|D|$ est le nombre de mots dans D, avgdl est la longueur moyenne des documents de $\mathcal{D}$. La fréquence inverse du mot q_i dans le corpus $\mathcal{D}$ est :

$$idf(q_i,\mathcal{D}) = \log\frac{|\mathcal{D}| - ndoc(q_i,\mathcal{D}) + 0.5}{ndoc(q_i,\mathcal{D}) + 0.5}$$

où $ndoc(q_i,\mathcal{D})$ est le nombre de documents contenant le mot q_i dans le corpus $\mathcal{D}$ (dont le nombre de documents est égal à $|\mathcal{D}|$). Deux variantes sont utilisées de manière similaire à TF-IDF.

6. telle qu'employée dans la librairie *Scikit-learn*
7. `https://radimrehurek.com/gensim/summarization/bm25.html`
8. Cette définition est reprise de l'article Wikipédia `https://fr.wikipedia.org/wiki/Okapi_BM25` cité par l'auteur de l'implémentation utilisée

Doc2Vec + similarité cosinus

Cette méthode est également basée sur le contenu et utilise la technique de représentation vectorielle Doc2Vec ou *Paragraph Vector Model* proposée par Le and Mikolov [2014]. L'objectif est d'obtenir des représentations vectorielles des articles à partir des titres et des abstracts. L'approche utilisée est similaire à Word2Vec à la différences que le réseau de neurones employé est également entraîné avec les documents complets (en l'occurrence les titres et les abstracts). L'idée est d'obtenir une représentation vectorielle du document qui soit la somme des représentations vectorielles des mots pondérées par leur importance. L'implémentation utilisée est celle de la librairie *Gensim*[9]. La taille des vecteurs est fixée à 100, le nombre d'époques (*epochs*) à 40 et le reste des paramètres à leur valeur par défaut comme proposé dans Cai, Han, Li, Zhang, Pan and Yang [2018]. Une fois les vecteurs obtenus, la similarité cosinus est utilisée pour comparer les articles. Deux variantes sont utilisées de manière similaire à TF-IDF.

PageRank personnalisé

Cette méthode [Haveliwala, 2003] est basée sur les graphes et utilise la variante personnalisée de PageRank [Page et al., 1999] appliquée au le graphe des citations. De manière similaire au PageRank original, l'idée est d'alterner les marches aléatoires dans le graphe des citations et les téléportations sur un nœud au hasard selon un certain paramètre, fixé ici à 0,85 (qui est la valeur par défaut de l'implémentation utilisée). La différence réside dans le fait que les marches aléatoires se font à partir d'une sélection de nœuds (i.e. les articles d'intérêt), de même que les téléportations. L'implémentation utilisée est celle de la librairie *NetworkX*[10]. Deux variantes sont utilisées respectivement sur les graphes orienté et non orienté des citations.

DeepWalk + similarité cosinus

Cette méthode est également basée sur les graphes et utilise la technique de représentation vectorielle DeepWalk [Perozzi et al., 2014] appliquée au le graphe des citations. Son principe de fonctionnement est similaire à Word2Vec qui est que le sens d'un mot est défini par son contexte, c'est-à-dire les autres mots à proximité. Le réseau de neurones utilisé pour obtenir les représentations des nœuds est entraîné sur des phrases qui sont en fait des marches aléatoires dans le graphe des citations. L'implémentation est inspirée de celle proposée par Perozzi et al. [2014] [11] et les paramètres utilisés sont [Ganguly and Pudi, 2017] : `number_walks=10, representation_size=64, seed=0, walk_length=40, window_size=5`. Une fois les vecteurs obtenus, la similarité cosinus est utilisée pour comparer les articles. Deux variantes sont utilisées respectivement sur les graphes orienté et non orienté des citations.

Node2Vec + similarité cosinus

Cette méthode est également basée sur les graphes et utilise la technique de représentation vectorielle Node2Vec [Grover and Leskovec, 2016] appliquée au le graphe des citations. Elle est une amélioration de DeepWalk en ce sens qu'elle permet de faire varier les marches aléatoires entre

9. `https://radimrehurek.com/gensim/models/doc2vec.html`
10. `https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_analysis.pagerank_alg.pagerank_scipy.html#networkx.algorithms.link_analysis.pagerank_alg.pagerank_scipy`
11. `https://github.com/phanein/deepwalk/`

profondeur et largeur. L'implémentation utilise le package Node2Vec [12] et les paramètres utilisés sont : `context_size_c = 10, dimensions_r = 128, walks_per_vertex T = 10, walk_length l = 80, p = 1, q = 2` comme suggérés par [Grover and Leskovec, 2016] pour découvrir les nœuds ayant des structures similaires et également utilisés par Cai et al. [2019]. Une fois les vecteurs obtenus, la similarité cosinus est utilisée pour comparer les articles. Deux variantes sont utilisées respectivement sur les graphes orienté et non orienté des citations.

SVD

Cette méthode est basée sur le filtrage collaboratif et utilise la technique de factorisation de matrice SVD [Koren et al., 2009] appliquée à la matrice d'adjacence du graphe orienté des citations. Ce qui équivaut à considérer chaque article comme un utilisateur dont le profil est caractérisé par ses références (i.e. sa bibliographie). L'implémentation utilisée est celle de la librairie *Scipy* [13].

Random

Cette méthode génère des recommandations aléatoires pour chaque entrée. Elle a principalement pour intérêt de vérifier que les autres méthodes apportent une réelle plus-value. Elle peut également indiquer la présence de bugs en mettant en évidence l'un ou l'autre comportement anormal.

5.2.2 Méthodes *state-of-art*

Le choix des méthodes *state-of-art* évaluées repose sur plusieurs critères. Le premier critère est de proposer une sélection représentative des différentes orientations de la recherche dans le domaine. Le deuxième critère est l'implémentabilité de la méthode. La disponibilité du code source est évidemment l'idéal. Cependant, une description suffisamment détaillée pour permettre l'implémentation et/ou la disponibilité des éléments constitutifs (par exemple les librairies) sont également acceptables. Le troisième critère important est la capacité de la méthode à répondre aux besoins de l'utilisateur. Une méthode candidate doit pouvoir prendre en compte d'autres objectifs que simplement la précision des recommandations. Idéalement, elle est paramétrable. Et elle doit bien sûr accepter efficacement plusieurs articles en entrée. Le dernier critère est relatif au corpus d'articles utilisé. Une méthode candidate doit être capable de gérer des jeux de données volumineux (plusieurs millions d'articles) tout en garantissant un temps de réponse acceptable, éventuellement moyennant des optimisations. De plus, elle ne doit pas nécessiter de données difficilement accessibles, comme les textes complets des articles. Enfin, elle peut facilement s'adapter à un corpus régulièrement augmenté et mis-à-jour.

Méthode 7

Cette méthode est proposée par Chakraborty et al. [2015]. Il s'agit d'une méthode hybride combinant graphe et contenu par combinaison de caractéristiques. Elle s'appuie sur le graphe orienté des citations et un graphe non-orienté et pondéré des mots-clés pour la sélection des candidats, et une variante de PageRank prenant en compte le prestige et la diversité pour le classement des candidats.

Le graphe non-orienté des mots-clés est un graphe où chaque nœud correspond à un mot-clé utilisé dans le corpus des items, et chaque arête entre deux mots-clés correspond à la présence conjointe

12. `https://github.com/eliorc/node2vec`
13. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.linalg.svds.html`

de ces deux mots-clés dans un même article. Chaque arête est associée à un poids qui équivaut au nombre d'articles où sont présents les mots-clés associés. Ce graphe est ensuite partitionné via la méthode de Louvain.

La variante de PageRank utilisée s'appelle *Vertex-Reinforced Random Walk* (VRRW). Elle se distingue par l'utilisation d'une matrice des probabilités de transition entre nœuds prenant en compte le nombre des visites précédentes, et qui évolue donc dans le temps.

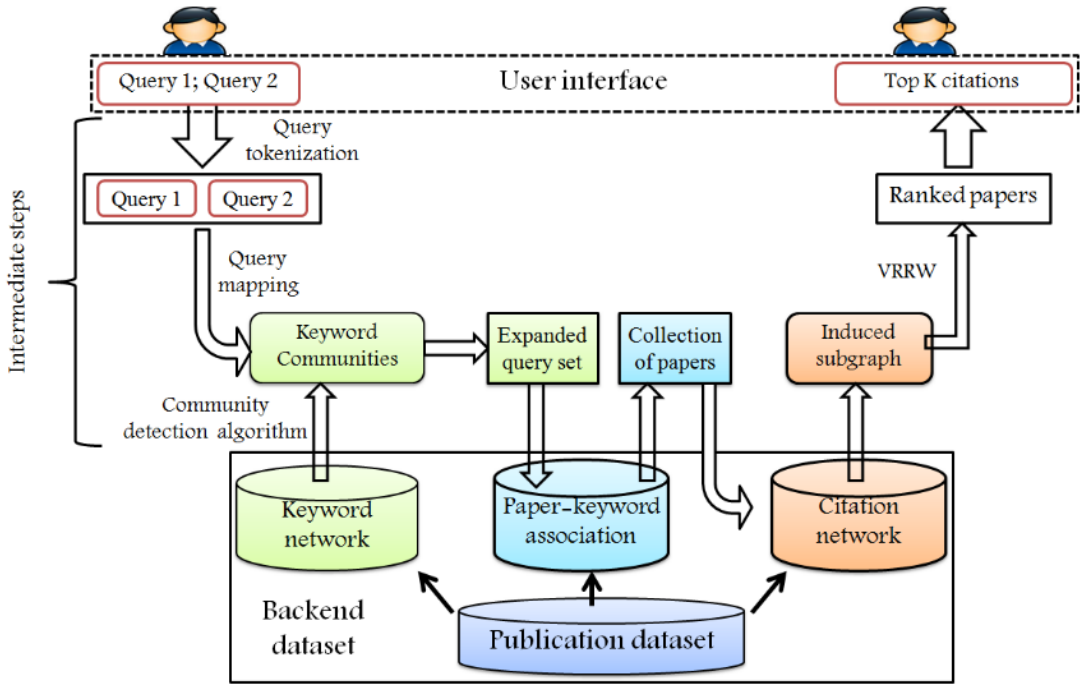

FIGURE 5.1 – Processus de recommandation de la méthode 7 (version locale) [Chakraborty et al., 2015].

Le processus de recommandation (voir figure 5.1) est le suivant :

1. L'ensemble des mots-clés d'intérêt Q est constitué à partir des articles sélectionnés par l'utilisateur.
2. Deux variantes : soit Q est utilisé directement (variante globale), soit $QExp$ est utilisé ($QExp$ est défini comme l'ensemble des mots-clés appartenant aux partitions des mots-clés de Q).
3. Tous les articles contenant au moins un mot-clé de Q (ou $QExp$) sont sélectionnés pour former l'ensemble des candidats,
4. Le sous-graphe orienté des citations est formé à partir des articles candidats.
5. Enfin, les candidats sont classés selon les scores de VRRW appliqué au sous-graphe des candidats.

Le choix de cette méthode est surtout motivé par l'utilisation des mots-clés, sa scalabilité et le fait qu'elle soit paramétrable.

Méthode 11

Cette méthode basée sur les graphes est proposée par West et al. [2016]. Elle combine l'utilisation d'une variante de PageRank, ALEF (pour *Article-Level Eigenfactor*), qui est optimisée pour les

graphes acyclique (comme le graphe des citations), et d'un partitionnement hiérarchique avec la méthode *MapEquation*[14]. Le processus général de recommandation est présenté dans la figure 5.2.

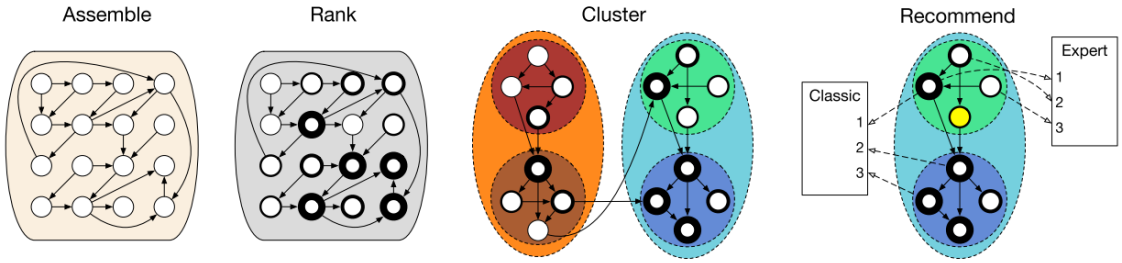

FIGURE 5.2 – Processus de recommandation de la méthode 11 (1. Construction du graphe orienté des citations, 2. calcul des scores ALEF, 3. partitionnement hiérarchique avec MapEquation, 4. recommandation selon le profil choisi) [West et al., 2016].

Tout d'abord, ALEF est appliquée au graphe orienté des citations pour obtenir les scores des n différents nœuds :

$$ALEF = n\frac{H_{ij}^T w_i}{\sum_j^n [H_{ij}^T w_i]_j}$$

avec H_{ij} la matrice d'adjacence normalisée par ligne, et w_i le vecteur de téléportation des nœuds i obtenu de la manière suivante : $w_i = \sum_j^n (Z_{ij} + Z_{ij}^T)$, Z_{ij} étant la matrice d'adjacence. Chaque nœud se voit attribuer un rang en fonction du score obtenu. Ensuite, le graphe est partitionné à partir des rangs en utilisant la méthode *MapEquation*. L'idée est donc d'obtenir un partitionnement hiérarchique qui soit représentatif des différents domaines et sous-domaines auxquels appartiennent les articles du graphe des citation. Et enfin, les recommandations sont calculées selon le profil choisi par l'utilisateur :

— *Expert* : sélection d'un cluster de dernier niveau et renvoi des meilleurs nœuds selon le classement calculé par ALEF,
— *Classique* : sélection d'un cluster d'avant-dernier niveau et renvoi des meilleurs nœuds selon le classement calculé par ALEF,
— *Sérendipité* : sélection d'un cluster de dernier niveau et choix aléatoire des nœuds.

Le premier intérêt de cette méthode est la faible complexité des opérations nécessaires à la recommandation proprement dite. En effet, la majorité des calculs (classement et partitionnement du corpus d'articles) peuvent être réalisés de manière asynchrone. Un second intérêt est la possibilité de paramétrer les recommandations en fonction du profil de l'utilisateur.

Méthode 20

Cette méthode appartient à la classe du filtrage collaboratif et est proposée par Haruna et al. [2017]. Le processus de recommandation est basé sur la recherche d'utilisateurs et d'articles similaires sur la matrice articles-citations. Cette matrice C est de taille $n \times n$ pour un corpus de n articles et pour chaque élément C_{ij}, $C_{ij} = 1$ si l'article i cite l'article j, $C_{ij} = 0$ sinon. Le processus de recommandation est le suivant (la méthode est adaptée pour pouvoir prendre plusieurs articles en entrée) :

1. Récupérer l'ensemble des références Rf_j des articles d'intérêt P_i
2. Construire l'ensemble P_{ci} avec les articles citant au moins un article de Rf_j

14. `https://www.mapequation.org/`

3. Récupérer l'ensemble des citations Cf_j des articles d'intérêt P_i
4. Construire l'ensemble P_{ri} avec les articles cités par au moins un article de Cf_j
5. Construire l'ensemble des candidats P_c avec les articles de P_{ci} et cités par au moins un article de P_{ri}
6. Le score de chaque candidat c appartenant à P_c est calculé de la manière suivante : $score(c) = \sum_i^{P_i} jaccard(C_c, C_i)$ avec $jaccard(C_c, C_i)$ l'indice de Jaccard entre les lignes c et i de la matrice C

Méthode 21 simplifiée

Cette méthode basée sur les graphes est proposée par Son and Kim [2018]. Elle utilise le graphe des citations et l'utilisateur est caractérisé par un article d'intérêt. Le processus de recommandation est constitué de 3 étape (voir également figure 5.3) :

1. Tout d'abord, un sous-graphe est construit à partir des nœuds à une distance maximale de 5 de l'article d'intérêt dans le graphe non orienté des citations. L'idée est bien sûr de limiter la complexité des calculs.
2. Ensuite, les 500 meilleurs candidats sont sélectionnés sur base de leur proximité sémantique avec le sujet d'intérêt. Cette proximité est déterminée en prenant en compte le couplage bibliographique et la co-citation entre les nœuds du sous-graphe et la distance par rapport à l'article d'intérêt. La fonction de score $score(i)$ pour un article candidat i est définie de la manière suivante :
$$score(i) = \frac{\sum_{j \in I \setminus \{i\}} cb(i,j) + cc(i,j)}{d(i,u)}$$
avec I l'ensemble des articles du sous-graphe, $cb(i,j)$ le nombre d'articles cités par i et j, $cc(i,j)$ le nombre d'articles citant i et j, et $d(i,u)$ la distance entre i et l'article d'intérêt u dans le graphe non orienté des citations.
3. Enfin, les candidats sont classés selon différentes mesures de centralité appliquées au sous-graphe construit à l'étape 1 : la centralité de degré, la centralité de proximité (*closeness centrality*), la centralité d'intermédiarité (*betweenness centrality*) et la centralité de vecteur propre. Le classement final est obtenu à partir du rang moyen.

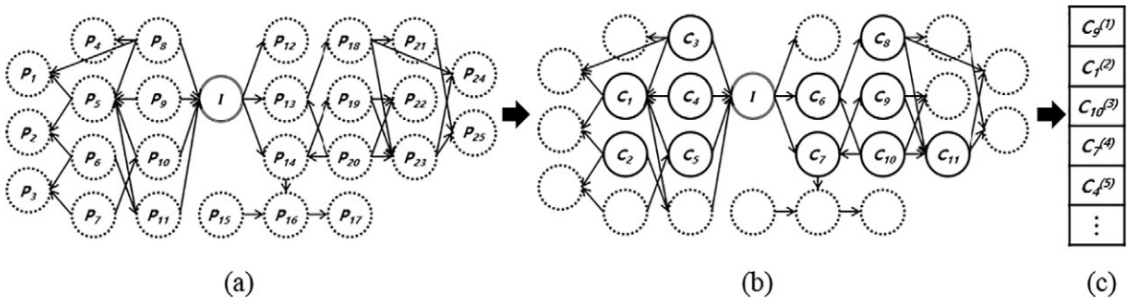

FIGURE 5.3 – Processus de recommandation de la méthode 21 (1. Construction du sous-graphe, 2. sélection des candidats, 3. Classement des candidats) [Son and Kim, 2018].

Bien que l'approche semble intéressante, cette méthode s'est avérée en pratique beaucoup trop lourde en temps de calcul. Elle a donc été adaptée pour diminuer le temps de calcul et également

pour pouvoir prendre en compte plusieurs articles d'intérêt. Le processus de recommandation est le suivant :

1. Un sous-graphe est construit à partir des nœuds à une distance maximale de 4 des articles d'intérêt dans le graphe non orienté des citations.
2. Pour chaque candidat i, le score est calculé à partir de l'indice de Jaccard avec les nœuds d'intérêt et de la centralité de vecteur propre.

$$score(i) = \frac{jaccard_u(i) + eigenvector(i)}{d_u(i)}$$

avec $jaccard_u(i)$ la moyenne des indices de Jaccard entre i et les articles d'intérêt, $eigenvector_u(i)$ la centralité de vecteur propre, et $d_u(i)$ la distance moyenne entre i et les articles d'intérêt. L'idée derrière cette fonction est de conserver la philosophie de la méthode initiale tout en simplifiant les temps de calcul.

Méthode 39

Cette méthode hybride combinant graphe et contenu par méta-niveau est proposée par Ganguly and Pudi [2017]. Elle combine les méthodes Doc2Vec et Node2Vec afin d'obtenir des représentations vectorielles des articles selon le processus suivant :

1. Un modèle Doc2Vec est entraîné à partir des titres et des abstracts des articles du corpus.
2. Un graphe non orienté des citations est construit. Pour chaque article, deux arêtes sont ajoutées vers les deux articles les plus similaires (selon leur représentation Doc2Vec).
3. Un modèle Node2Vec est entraîné à partir du graphe construit à l'étape 2. Les poids entre les entrées et la couche cachée du modèle sont au préalable initialisés avec les poids du modèle Doc2Vec entraîné à l'étape 1.
4. Ce dernier modèle fournit les représentations finales des articles du corpus.

Enfin, les recommandations sont obtenues par similarité cosinus avec les articles d'intérêt.

Cette méthode a été choisie principalement pour avoir une méthode d'*embeddings* intégrant contenu et graphe, et parce que le code source était disponible (ce qui permet d'avoir tous les paramètres utilisés pour les différents modèles).

5.3 Analyse des résultats

Liste des méthodes comparées :

— `base_tfidfcosine_sum` : TF-IDF + similarité cosinus, somme des scores de similarité.
— `base_tfidfcosine_concat` : TF-IDF + similarité cosinus, concaténation des titres et des abstracts.
— `base_bm25_sum` : BM25, somme des scores de similarité.
— `base_bm25_concat` : BM25, concaténation des titres et des abstracts.
— `base_doc2veccosine_sum` : Doc2Vec + similarité cosinus, somme des scores de similarité.
— `base_doc2veccosine_concat` : Doc2Vec + similarité cosinus, concaténation des titres et des abstracts.
— `base_personalizedpagerank_directed` : PageRank personnalisé, graphe orienté des citations.

— `base_personalizedpagerank_undirected` : PageRank personnalisé, graphe non-orienté des citations.
— `base_deepwalkcosine_directed` : DeepWalk + similarité cosinus, graphe orienté des citations.
— `base_deepwalkcosine_undirected` : DeepWalk + similarité cosinus, graphe non-orienté des citations.
— `base_node2veccosine_directed` : Node2Vec + similarité cosinus, graphe orienté des citations.
— `base_node2veccosine_undirected` : Node2Vec + similarité cosinus, graphe non-orienté des citations.
— `base_svd` : factorisation de la matrice des citations (SVD).
— `base_random` : recommandations aléatoire.
— `stateofart_method7_glo` : méthode 7, variante globale (pas d'ajout de mots-clés).
— `stateofart_method7_loc` : méthode 7, variante locale (ajout de mots-clés).
— `stateofart_method11_classic` : méthode 11, variante *classique* (cluster d'avant-dernier niveau + ALEF).
— `stateofart_method11_expert` : méthode 11, variante *expert* (cluster de dernier niveau + ALEF).
— `stateofart_method11_serendipity` : méthode 11, variante *sérendipité* (cluster de dernier niveau + sélection aléatoire).
— `stateofart_method20` : méthode 20.
— `stateofart_method21simplified` : méthode 21 simplifiée.
— `stateofart_method39` : méthode 39.

Afin de présenter les résultats, la visualisation employée exploite les résultats du test de Nemenyi et consiste à positionner les différentes méthodes selon leur rang sur une échelle graduée. Des traits gras sont ensuite ajoutés lorsque la différence critique entre deux méthodes n'est pas significative. Par exemple, dans la figure 5.4, `base_tfidfcosine_sum` est légèrement meilleure que `base_tfidfcosine_concat` mais la présence d'un trait gras indique que cette différence n'est pas significative.

5.3.1 Précision

Contenu

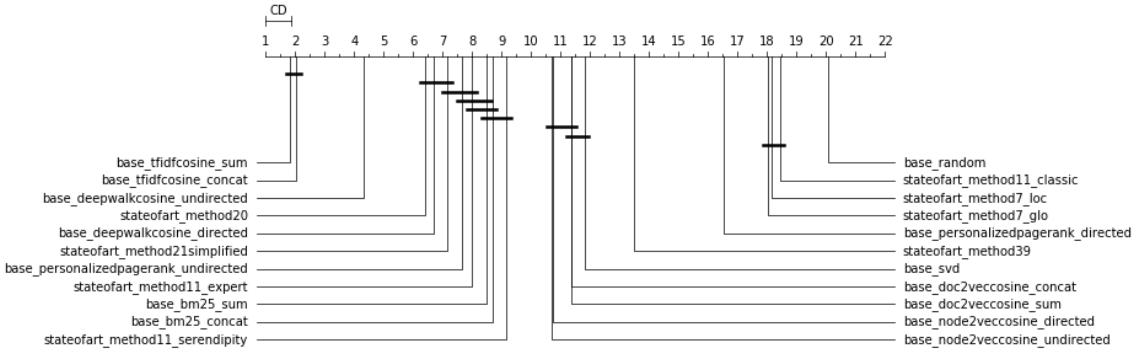

FIGURE 5.4 – Résultats pour la précision selon le contenu (AAN).

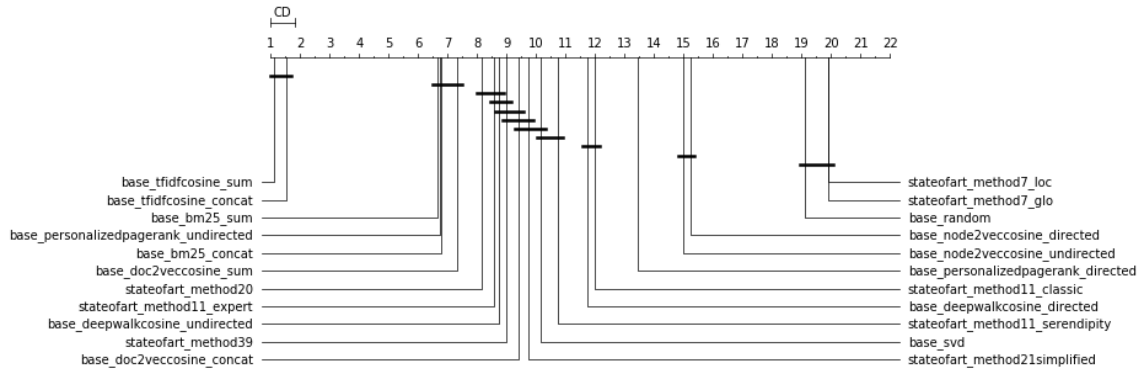

FIGURE 5.5 – Résultats pour la précision selon le contenu (DBLP).

Le premier constat des résultats obtenus (voir figures 5.4 et 5.5) est la supériorité des méthodes TF-IDF. Ce qui confirme le fait que TF-IDF reste une méthode tout à fait efficace pour estimer la similarité sémantique à partir du moment où les textes analysés ne se limitent pas à une ou deux phrases [Alvarez and Bast, 2017, Shahmirzadi et al., 2019]. Par contre, la moins bonne performance des méthodes BM25 est plus étonnante car la littérature en recherche d'information tend à montrer qu'elle se situe plus ou moins au même niveau que TF-IDF, voire légèrement au dessus.

Par contre, les deux variantes de la méthode 7 semblent ne pas fonctionner du tout. Il semblerait que le problème provienne de l'indexation des articles. Certains mots-clés sont en effet employés pour une grande partie du corpus, avec pour conséquence un ensemble de candidats trop important et donc des recommandations pas assez spécifiques.

Concernant les méthodes basées sur les graphes, les variantes utilisant un graphe non orienté semblent meilleures que leurs homologues utilisant un graphe orienté. Un résultat surprenant est que DeepWalk semble meilleure que Node2Vec. Pourtant, Node2Vec est sensé être une amélioration de DeepWalk. Une explication pourrait être le choix des paramètres qui n'est pas optimal. Enfin, il y a pas mal de variabilité entre les résultats selon le jeu de données utilisé. Néanmoins, les méthodes *state-of-art* ne semblent pas plus performantes que les méthodes de base.

Références et citations

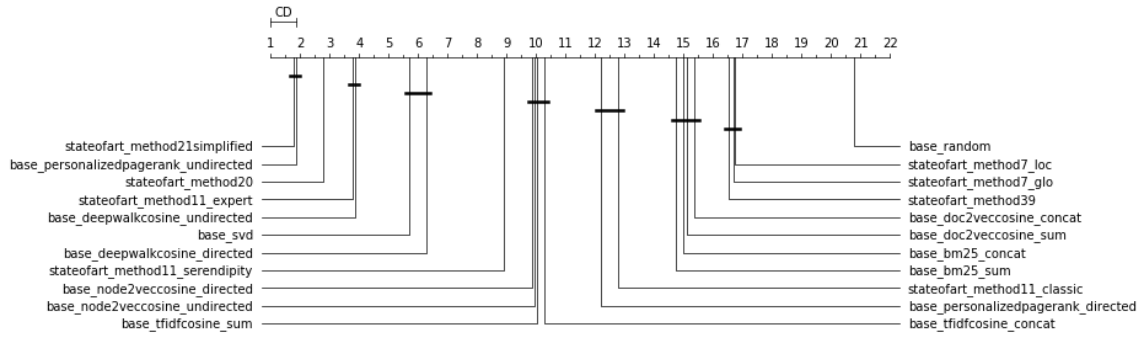

FIGURE 5.6 – Résultats pour la précision selon les références et les citations (AAN).

De manière générale, les résultats (voir figures 5.6 et 5.7) montrent une prédominance des méthodes basées sur les graphes. Ce qui est évidemment cohérent avec la mesure utilisée. De plus,

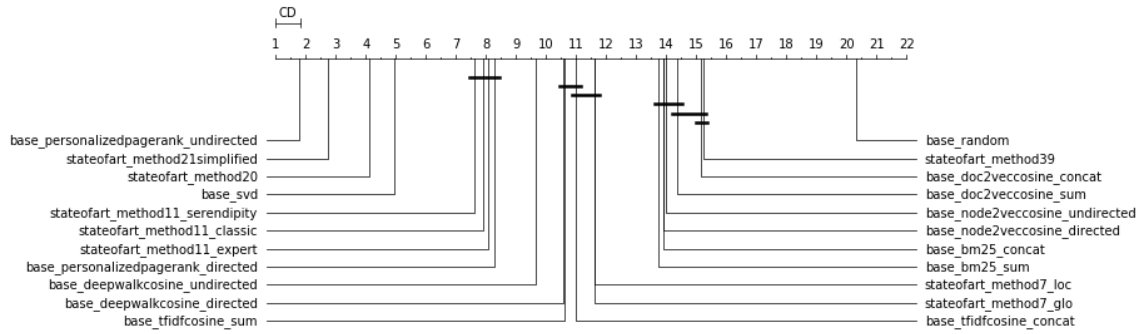

FIGURE 5.7 – Résultats pour la précision selon les références et les citations (DBLP).

les variantes basées sur un graphe non orienté semblent plus performantes que leurs homologues sur un graphe orienté. La mesure utilisée prenant en compte les références et les citations des articles, ce constat est également tout à fait cohérent. Parmi les autres méthodes, TF-IDF semble se démarquer légèrement. Enfin, les performances de la méthode 39 ne sont pas très bonnes par rapport à des méthodes proches comme Node2Vec et DeepWalk. Le problème pourrait être lié au choix des paramètres qui n'est pas optimal, notamment le nombre de dimensions des vecteurs qui est peut-être trop faible.

5.3.2 Diversité

Contenu

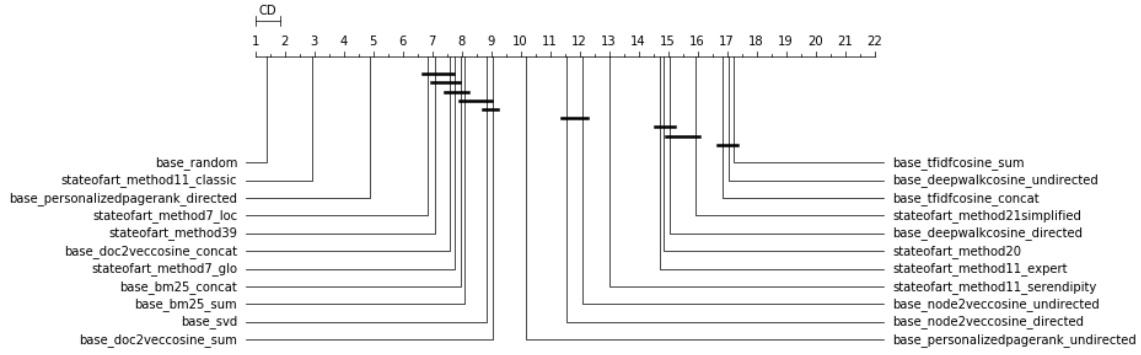

FIGURE 5.8 – Résultats pour la diversité selon le contenu (AAN).

Les résultats (voir figures 5.8 et 5.9) sont assez difficiles à interpréter étant donné les nombreuses différents entre AAN et DBLP. Néanmoins, les classements semblent suivre une tendance inverse à ceux de la précision de contenu, ce qui est notamment le cas des méthodes 7 et TF-IDF.

Références et citations

Le constat avec ces résultats (voir figures 5.10 et 5.11) est similaire à la diversité de contenu. Malgré la forte variabilité entre AAN et DBLP, les classements semblent suivre une tendance inverse, bien que moins marquée ici, par rapport à ceux de la précision des références et citations.

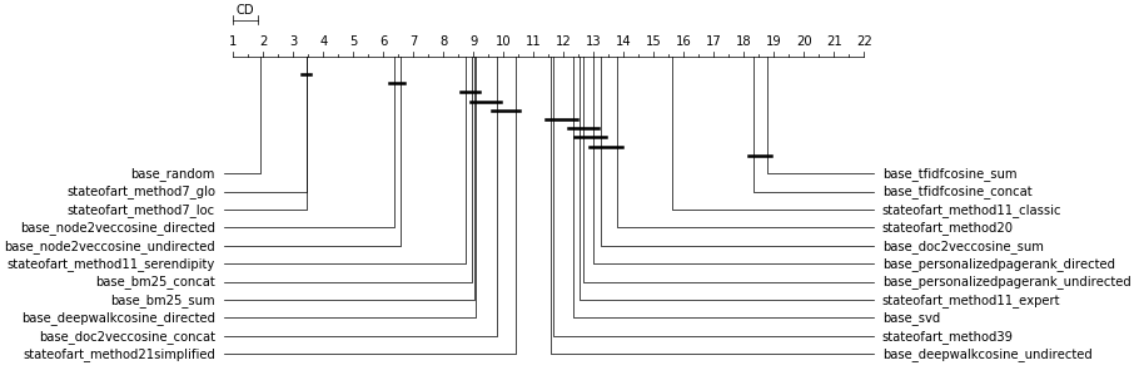

FIGURE 5.9 – Résultats pour la diversité selon le contenu (DBLP).

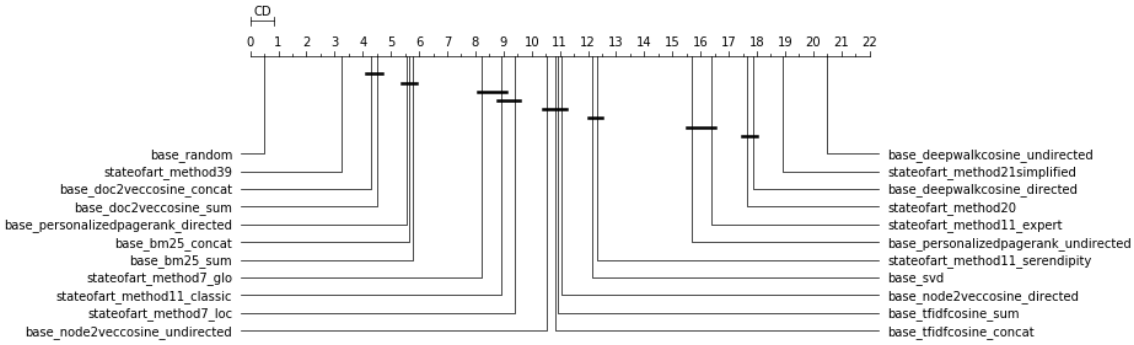

FIGURE 5.10 – Résultats pour la diversité selon les références et les citations (AAN).

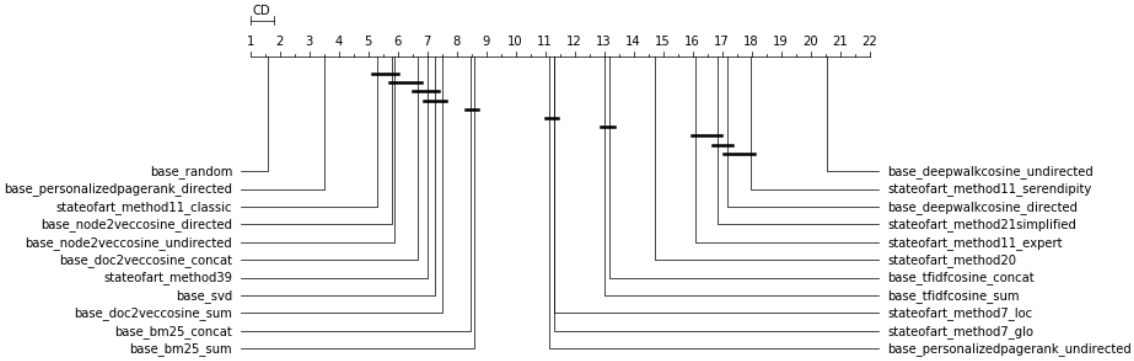

FIGURE 5.11 – Résultats pour la diversité selon les références et les citations (DBLP).

5.3.3 Nouveauté

Date de publication

Les résultats (voir figures 5.12 et 5.13) montrent clairement la supériorité de la méthode 7. L'explication la plus probable est sans doute la méthode de classement (ALEF) qui favorise notamment cet aspect [Chakraborty et al., 2015]. Les méthodes aléatoires (`base_random` et `stateofart_method11_serendipity`) sont également assez bonnes ici. Par contre, les méthodes basées sur les graphes semblent être globalement moins performantes. Une explication pourrait être le plus faible nombre de citations pour

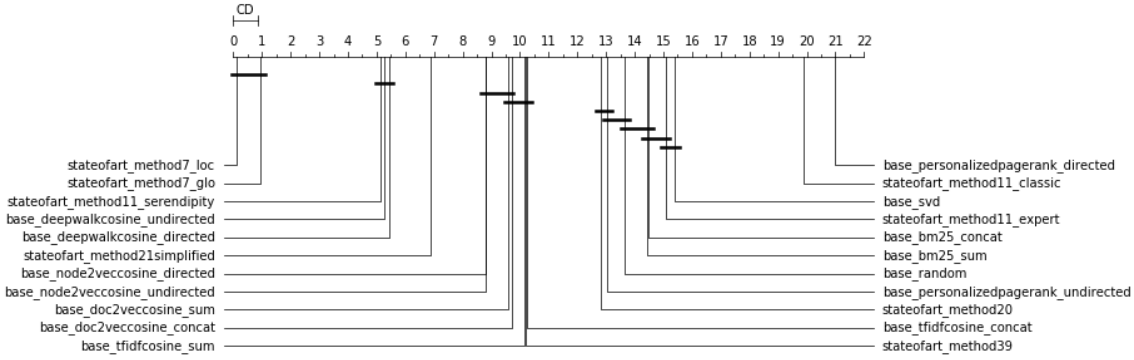

FIGURE 5.12 – Résultats pour la nouveauté selon l'année de publication (AAN).

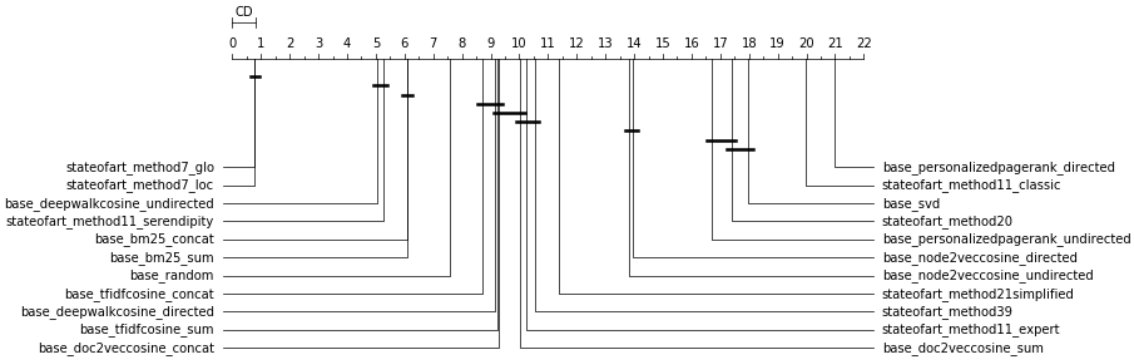

FIGURE 5.13 – Résultats pour la nouveauté selon l'année de publication (DBLP).

les articles récents, et donc de connexions dans le graphe des citations.

Popularité inverse

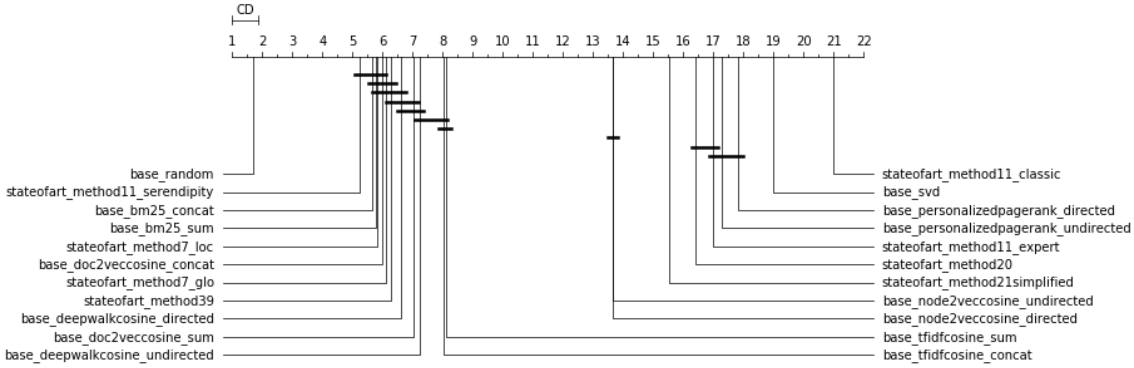

FIGURE 5.14 – Résultats pour la nouveauté selon la popularité inverse (AAN).

Les résultats (voir figures 5.14 et 5.15) sont également difficiles à interpréter étant donné les nombreuses différences entre AAN et DBLP. Néanmoins, les méthodes basées sur les graphes semblent moins performantes que les autres méthodes. Une explication pourrait être l'existence d'un lien entre le nombre de relations de citation des candidats et le score d'évaluation prédit. Enfin, les

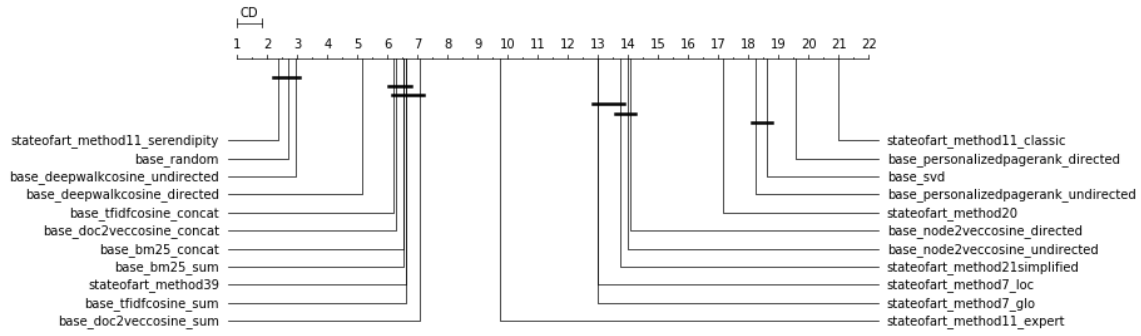

FIGURE 5.15 – Résultats pour la nouveauté selon la popularité inverse (DBLP).

méthodes aléatoires (`base_random` et `stateofart_method11_serendipity`) sont assez bonnes ici aussi.

5.3.4 Couverture

Items

De manière générale, la couverture d'items (voir table 5.1) est bonne pour la plupart des méthodes avec un taux supérieur à 60 %. À noter la forte différente entre les variantes orientée et non orientée du Pagerank personnalisé. Les méthodes basées sur le filtrage collaboratif (`base_svd` et `stateofart_method20`) ont également un taux de couverture assez faible. Enfin, le taux de couverture extrêmement faible de la méthode 7 et de la variante classique de la méthode 11 montre que ces méthodes ont tendance à recommander les mêmes articles. Ce phénomène est dû à la combinaison d'un nombre de candidats sélectionnés trop large et l'utilisation d'une fonction de scoring qui n'est pas personnalisée.

Utilisateurs

Comme le montre la table 5.1, la couverture utilisateurs est totale pour la plupart des méthodes à l'exception de quatre. Dans le cas des variantes expert et sérendipité de la méthode 11, la cause est sans doute la taille du cluster de candidats qui est parfois trop faible. Dans le cas de la méthode 20, la sélection des candidats est basée sur les relations directes de citation entre les candidats et les articles d'intérêt, avec pour conséquence un nombre trop faible de candidats pris en compte pour une partie des entrées. Le problème est similaire pour la méthode 21, mais dans une moindre mesure car les citations indirectes (jusqu'à une distance de 4) sont prises en compte.

TABLE 5.1 – Couvertures items et utilisateurs.

Méthode	Items (AAN)	Items (DBLP)	Users (AAN)	Users (DBLP)
base_bm25_concat	67,60 %	60,86 %	100,00 %	100,00 %
base_bm25_sum	67,95 %	61,06 %	100,00 %	100,00 %
base_deepwalkcosine_directed	94,55 %	86,72 %	100,00 %	100,00 %
base_deepwalkcosine_undirected	96,98 %	91,66 %	100,00 %	100,00 %
base_doc2veccosine_concat	99,85 %	94,68 %	100,00 %	100,00 %
base_doc2veccosine_sum	96,59 %	87,63 %	100,00 %	100,00 %
base_node2veccosine_directed	63,25 %	38,29 %	100,00 %	100,00 %
base_node2veccosine_undirected	63,30 %	38,15 %	100,00 %	100,00 %
base_personalizedpagerank_directed	37,49 %	12,78 %	100,00 %	100,00 %
base_personalizedpagerank_undirected	92,02 %	74,32 %	100,00 %	100,00 %
base_random	100,00 %	99,97 %	100,00 %	100,00 %
base_svd	29,32 %	15,52 %	100,00 %	100,00 %
base_tfidfcosine_concat	92,23 %	81,41 %	100,00 %	100,00 %
base_tfidfcosine_sum	91,15 %	80,01 %	100,00 %	100,00 %
stateofart_method11_classic	0,84 %	0,45 %	100,00 %	100,00 %
stateofart_method11_expert	68,55 %	69,22 %	81,00 %	83,70 %
stateofart_method11_serendipity	91,64 %	84,46 %	81,80 %	84,75 %
stateofart_method20	38,96 %	18,33 %	65,25 %	76,95 %
stateofart_method21simplified	78,12 %	67,89 %	98,05 %	97,50 %
stateofart_method39	93,76 %	84,39 %	100,00 %	100,00 %
stateofart_method7_glo	0,93 %	0,45 %	100,00 %	100,00 %
stateofart_method7_loc	0,93 %	0,45 %	100,00 %	100,00 %

Chapitre 6

Conception et réalisation du prototype

Ce chapitre est consacré à la conception et à la réalisation du prototype. Il commence par présenter les choix de conception et les justifier à partir de l'analyse et des résultats de la comparaison des méthodes de recommandation. Ensuite, le principe de fonctionnement de l'outil est détaillé du point de vue de l'utilisateur. Enfin, les différents aspects de l'implémentation sont évoqués. Le code source du prototype est disponible dans l'annexe A.6 et une vidéo de présentation est proposée dans l'annexe A.7. Enfin, le prototype est accessible à l'adresse suivante : `https://stage-bibliographie.cetic.be/`.

6.1 Conception de l'outil

Initialement, l'idée était d'exploiter les résultats de l'évaluation hors-ligne en sélectionnant quelques méthodes ayant en commun une bonne précision de contenu mais avec des performances dans les autres aspects évalués davantage différentes. Ces méthodes auraient été proposées à l'utilisateur, en mettant avant leurs qualités, et ce dernier aurait pu choisir la plus adaptée en fonction de ses besoins. Cependant, force est de constater que les résultats obtenus ne permettent pas de conserver cette approche. En effet, comme l'analyse l'a montré (voir chapitre 4), la précision des recommandations est primordiale pour la plupart des personnes interrogées. Certaines ont d'ailleurs une mauvaise perception des systèmes de recommandation justement à cause du manque de précision. Et donc, étant donné la supériorité des méthodes TF-IDF sur les autres méthodes en terme de précision de contenu, le choix d'une méthode TF-IDF s'impose même si elle est moins performante selon d'autres aspects.

Malgré tout, le système doit également intégrer d'autres dimensions comme la nouveauté ou la diversité dans ses recommandations. Pour ce faire, une stratégie de reclassement des recommandations, inspirée de la recommandation multi-objectifs (voir section 2.5), est appliquée. Le principe est de générer un ensemble suffisamment riche de recommandations avec un bon niveau de précision et de permettre à l'utilisateur de reclasser les résultats selon ses besoins. Du point de vue du développement, cette stratégie a l'avantage de simplifier grandement le travail d'implémentation par rapport à la stratégie initialement prévue. De plus, les optimisations nécessaires en cas de passage en production, avec notamment un corpus d'articles plus conséquent, seront également facilitées.

Les critères de tri sont inspirés des mesures employées dans l'évaluation hors-ligne (voir section 3.4) et leur interprétation est d'ailleurs identique. Le premier critère est la précision, exprimée comme la similarité entre les recommandations et les articles sélectionnés par l'utilisateur. Cette similarité est décliné en deux variantes : la similarité textuelle (en prenant en compte le titre et l'abstract) et la similarité topologique (i.e. les références et citations communes).

Le deuxième critère est la diversité. Il cherche à favoriser la présence en début de liste d'une sélection d'articles aussi variée que possible. L'objectif est d'aider à l'utilisateur d'avoir une vue globale du sujet exploré. Ce critère est également décliné en deux variantes en fonction des données prises en compte. La première variante utilise les titres et les abstracts et favorise donc la diversité de contenu. La seconde utilise les références et les citations et favorise plutôt la diversité topologique.

Enfin, le troisième critère envisagé est la nouveauté. Le but est de favoriser en début de liste la présence d'articles potentiellement inconnus pour l'utilisateur. La première variante utilise la date de publication et cherche donc à mettre en avant les articles récents. La seconde variante se base sur la popularité inverse, approximée par le nombre de citations, et favorise en début de liste les articles moins connus des chercheurs en général.

Enfin, ces critères peuvent être combinés via un système de pondération. Ce choix est motivé par deux raisons. Premièrement, ce système de pondération permet d'utiliser des valeurs négatives et donc d'inverser la signification des critères. Ceci permet d'augmenter les possibilités de tri. Par exemple, utiliser une valeur négative pour la popularité inverse permet d'ordonner les articles selon leur popularité et donc de favoriser la présence d'articles de référence dans les premiers résultats. Un autre exemple est l'utilisation d'une valeur négative pour la similarité topologique qui est une approche très complémentaire d'une première recherche par exploration des bibliographies. Deuxièmement, la possibilité de combiner les critères augmente la finesse de ce qui peut être exprimé dans le système. Par exemple, combiner la similarité de contenu avec une valeur positive et la similarité topologique avec une valeur négative peut être une bonne manière d'identifier rapidement les recherches similaires mais effectuées par des groupes de chercheurs différents.

6.2 Principe de fonctionnement et interface

6.2.1 Principe général

Le système de recommandation proposé est conçu comme un véritable outil de recherche bibliographique, avec pour ambition d'être à terme une véritable alternative aux outils traditionnels comme les moteurs de recherche spécialisés. Cependant, il s'agit ici d'un prototype dont le but est surtout de valider la stratégie de recherche bibliographique proposée ici. Son utilisation repose sur un cycle de recommandation (voir figure 6.1) structuré en trois étapes :

1. la sélection d'articles d'intérêt via l'importation de références bibliographiques au format BibTeX (module LOAD) et la recherche textuelle (module SEARCH),
2. la recommandation à partir des articles sélectionnés et de la méthode choisie (module RECOMMEND),
3. et l'exploitation des résultats avec la consultation et les différentes possibilités de tri (module EXPLOIT), la mise en évidence des auteurs, organes de publication et mots-clés importants (module DASHBOARD), et l'exportation des articles sauvegardés au format BibTeX (module SAVE).

Étant donné que la recherche bibliographique est un processus souvent itératif, l'outil permet également que les résultats du cycle de recommandation précédent puissent alimenter le cycle suivant. Enfin, une page d'accueil présente l'outil et son fonctionnement.

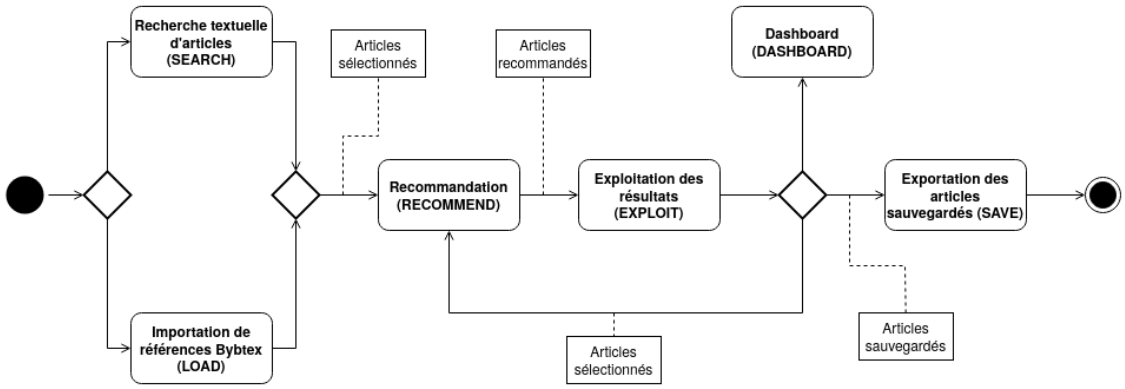

FIGURE 6.1 – Diagramme présentant les 3 phases de l'outil et mettant en avant les principes de recherche itérative et de cycle de recherche.

6.2.2 Conception de l'interface

La conception de l'interface n'a pas été l'objet d'une recherche spécifique. L'interface proposée est inspirée de divers outils de recherche bibliographique, et avec la volonté de proposer quelque chose de simple et de relativement intuitif. Elle est implémentée avec le framework *Bootstrap*[1] et à partir du template *Dashboard*[2].

La structure type d'une page est composée de trois parties (voir figure 6.2 pour un exemple) :

— un menu général positionné horizontalement au dessus et qui permet de naviguer entre les différents modules,

— un menu contextuel positionné verticalement à gauche et qui contient les options spécifiques aux différents modules,

— et le contenu des différents modules.

Pour chaque module dont le fonctionnement ou les fonctionnalités méritant quelques explications, un encart explicatif est présent en dessous du titre. Celui-ci peut être caché via le bouton `Hide/Show`. Le choix de le laisser apparent par défaut est motivé par les premiers retours qui n'ont pas toujours remarquer celui-ci via la simple présence du bouton. Ce choix pourrait sans doute être revu par la suite.

L'usage des couleurs est limité aux informations importantes et aux actions. Un code couleur est utilisé pour les boutons. Le bleu est utilisé par défaut et pour l'ajout de références en entrée pour la recommandation, le vert pour la sauvegarde des références, le rouge pour la suppression et le turquoise pour l'aide. Un autre code couleur est utilisé pour les alertes : jaune pour les avertissements, rouge pour les erreurs et vert pour les succès.

Enfin, les références bibliographiques sont présentées de manière tabulaire toujours par souci de clarté. L'intégralité des méta-données sont accessibles par expansion de la ligne correspondante. Les tableaux sont implémentés avec l'extension *Boostrap Table*[3].

1. `https://getbootstrap.com/`
2. `https://getbootstrap.com/docs/4.4/examples/dashboard/`
3. `https://bootstrap-table.com/`

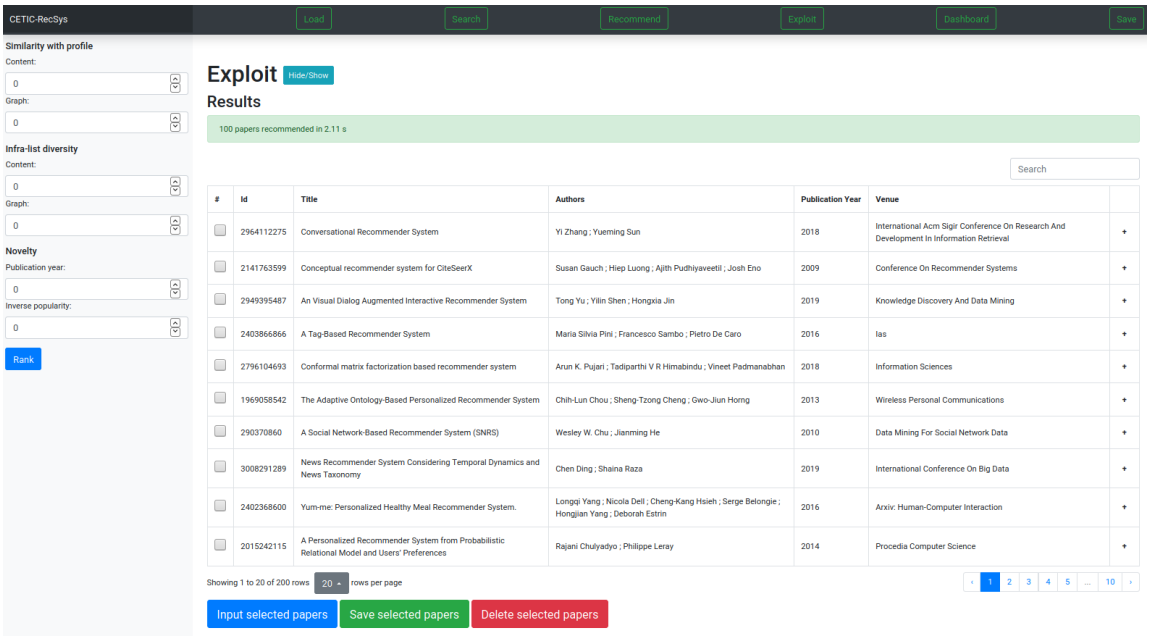

FIGURE 6.2 – Capture d'écran du prototype montrant les différents éléments d'une page type.

6.3 Implémentation

6.3.1 Architecture et choix technologiques

Les seules exigences pour la réalisation du prototype étaient qu'il devait prendre la forme d'une application web et d'éviter les technologies « exotiques ». Le prototype ne devait notamment pas passer à l'échelle en l'état même si cette aspect devait rester à l'esprit lors de la conception. Le choix de Python comme langage de développement s'est rapidement imposé pour de nombreuses raisons liées à la richesse de l'écosystème (librairies, documentation, communauté, etc.), son omniprésence en data science, le fait qu'il soit adapté au présent projet, et son utilisation fréquente au CETIC.

Le développement de l'application web proprement dite est réalisé avec le framework *Flask*[4]. Il s'agit d'un framework web très populaire, notamment pour le développement de petites applications. Son intérêt principal est qu'il est assez léger et peut être mis en œuvre assez rapidement. De nombreuses extensions sont également disponibles en fonction des besoins spécifiques de l'application développée. Il convient donc plutôt bien au prototype étant donné la relative simplicité de celui-ci.

Arborescence des fichiers principaux constituant le prototype

- `static/` : fichiers `*.css` et `*.js`,
- `templates/` : fichiers `*.html`,
- `__init__.py` : création de l'application avec les modules associés,
- `config.py` : paramètres de l'application et de connexion aux bases de données,
- `dashboard.py` : module DASHBOARD,
- `db.py` : accès aux bases de données,
- `exploit.py` : module EXPLOIT,

4. `https://flask.palletsprojects.com/`

— `index.py` : module de la page d'accueil,
— `load.py` : module LOAD,
— `rank.py` : fonctions pour reclassement des recommandations,
— `recommend.py` : module RECOMMEND,
— `recommend_methods.py` : méthodes de recommandation,
— `save.py` : module SAVE,
— `search.py` : module SEARCH,
— `utilities.py` : fonctions utilitaires (principalement pour le formatage des références bibliographiques et la gestion des variables de session).

6.3.2 Gestion des données

Le corpus d'items utilisé par le prototype provient du jeu de données *DBLP-Citation-network*[5] dans sa version 12 et contenant près de 5 millions d'articles. Le choix d'un corpus fermé est surtout motivé par la facilité de mise en œuvre. En effet, l'implémentation d'un mécanisme de recherche de références bibliographiques via des services web tiers représente un travail conséquent sans pour autant être indispensable dans le cadre d'un premier prototype. Ce choix a néanmoins pour conséquence que la recherche de références bibliographiques spécifiques peut parfois échouer (problème qui concerne les modules LOAD et SEARCH).

Concrètement, la gestion des données est répartie en deux composants : une base de données relationnelle pour la gestion des requêtes classiques de lecture, et une solution spécifique pour les requêtes impliquant une recherche textuelle.

PostgreSQL

Étant donné le volume encore raisonnable de données, l'idée de départ était de normaliser le schéma en distinguant notamment les auteurs, les mots-clés et les organes de publications, et donc d'utiliser une base de données relationnelle permettant d'exprimer facilement ces contraintes. Le choix s'est rapidement porté sur PostgreSQL[6] pour plusieurs raisons. Il s'agit d'un système de gestion de base de données libre, avec des pilotes Python, ayant une bonne compatibilité avec la norme SQL, et ses performances en lecture sont tout à fait acceptables.

Cependant, des problèmes de performance ont rapidement été constatés avec des requêtes impliquant des jointures. Ce qui a conduit à dénormaliser certaines relations en intégrant les clés étrangères dans les tables principales. Ce choix est également motivé par le fait qu'il n'y ait que des requêtes de lecture dans le prototype, ce qui n'implique pas de vérification des contraintes du schéma après le chargement initial des données. Rétrospectivement, une base de données orientée document comme MongoDB aurait pu également convenir. Le schéma final est présent sur la figure 6.3.

Elasticsearch

La recherche textuelle dans un corpus aussi important nécessite forcément une solution spécifique. Plusieurs possibilités étaient envisageables : *Apache Lucene*[7], *Apache Solr*[8] et *Elasticsearch*[9].

5. `https://www.aminer.org/citation`
6. `https://www.postgresql.org/`
7. `https://lucene.apache.org/`
8. `https://lucene.apache.org/solr/`
9. `https://www.elastic.co/`

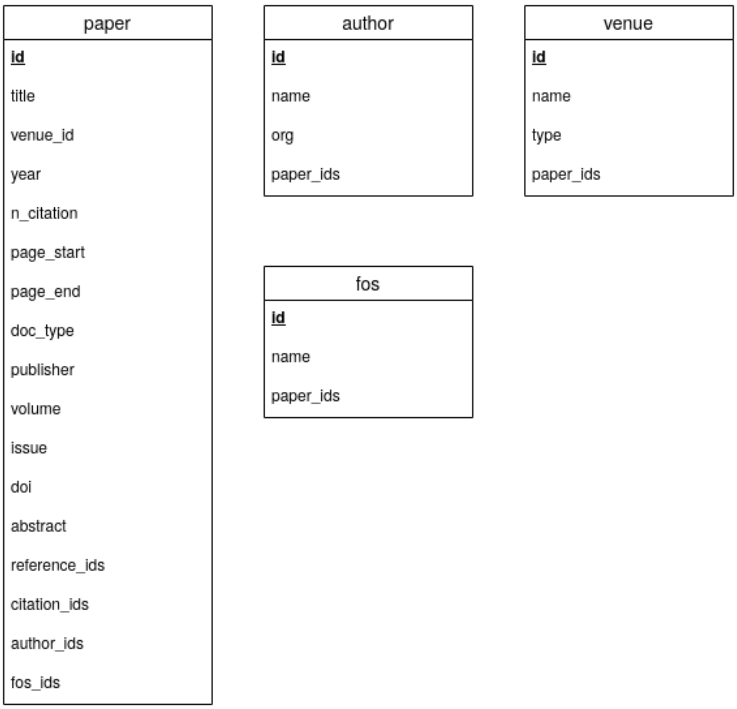

FIGURE 6.3 – Schéma entité-association final de la base de données PostgreSQL.

Le choix s'est finalement porté sur *Elasticsearch* pour les raisons suivantes : la popularité de cette solution, la facilité de mise en œuvre, la richesse de la documentation, la présence d'un client pour Python, de nombreuses possibilités de paramétrage et fonctionnalités *built-in*, la scalabilité, etc.

Un des intérêts principaux d'*Elasticsearch* est la fonction `more_like_this` qui permet de rechercher des documents similaires à un ou plusieurs documents passés en paramètre. Cependant, *Elasticsearch* utilise par défaut BM25 comme fonction de similarité pour les données textuelles. Étant donné les résultats de la sélection des méthodes, il était donc nécessaire d'utiliser une fonction basée sur TF-IDF. Celle-ci a été implémentée par l'intermédiaire d'une fonction de similarité scriptée [10]. Le code de la fonction et le schéma final sont présentés dans les listings 6.1 et 6.2

Listing 6.1 – Code de la fonction de similarité TF-IDF.

```
"similarity": {
 "scripted_tfidf": {
  "type": "scripted",
  "script": {
   "source": "double tf = Math.sqrt(doc.freq);
    double idf = Math.log((field.docCount+1.0)/(term.docFreq+1.0)) +
        1.0;
    double norm = 1/Math.sqrt(doc.length);
    return query.boost * tf * idf * norm;"
  }
 }
}
```

10. https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules-similarity.html

Listing 6.2 – Schéma *Elasticsearch* final.

```
{
 "dblp_v12_v2" : {
 "mappings" : {
  "properties" : {
   "abstract" : { "type" : "text", "similarity" : "scripted_tfidf" }
       ,
   "doi" : {
    "type" : "text",
    "fields" : { "keyword" : { "type" : "keyword", "ignore_above" :
       256 } }
   },
   "fos" : {
    "type" : "text",
    "fields" : { "keyword" : { "type" : "keyword", "ignore_above" :
       256 } }
    },
   "linked_papers" : { "type" : "long" },
   "paper_id" : { "type" : "long" },
   "title" : { "type" : "text", "similarity" : "scripted_tfidf" }
   }
  }
 }
}
```

6.3.3 Aspects fonctionnels

Le fonctionnement général de l'application repose sur l'utilisation de variables de session. Une variable de session est initialisée lors de la première visite d'un utilisateur et elle conserve les derniers résultats retournés par les différents modules. Elle conserve également la dernière pondération utilisée pour le reclassement.

LOAD

LOAD est le premier module de la phase de sélection des articles d'intérêt. Son but est de permettre de charger des articles à partir de références bibliographiques au format BibTeX. Il s'agit d'ailleurs de sa méthode principale. Dans un premier temps, celle-ci tente parser les données BibTeX fournie via le formulaire à l'aide de la librairie *Pybtex*[11]. Ensuite, pour chaque enregistrement parsé, elle récupère le document le plus proche dans la base *Elasticsearch* (avec comme ordre de priorité : 1. DOI, 2. titre et 3. abstract). Enfin, elle retourne la liste à l'utilisateur.

Listing 6.3 – Requête de recherche.

```
{
 'size': 1,
```

11. https://pybtex.org/

```
  'query': {
   'multi_match': {
   'query': query,
    'type': 'best_fields',
    'fields': ['title^3', 'abstract', 'doi^5']
   }
  }
}
```

Routes associées :

- `/load` : chargement de données BibTeX, recherche des documents correspondants et renvoi des résultats,
- `/add_from_load` : ajout des articles sélectionnés aux entrées pour recommandation,
- `/save_from_load` : sauvegarde des articles sélectionnés dans le module SAVE,
- `/delete_from_load` : suppression des articles sélectionnés dans la liste du module LOAD.

SEARCH

SEARCH est le second module de la phase de sélection des articles d'intérêt. Son but est de permettre la recherche textuelle d'articles. Sa méthode principale recherche les 10 documents les plus proches de la requête textuelle dans la base *Elasticsearch* (requête identique à celle employée dans le module LOAD à l'exception du paramètre `size` qui est ici à 10, voir listing 6.3) et retourne la liste des résultats.

Routes associées :

- `/search` : recherche des documents correspondants et renvoi des résultats,
- `/add_from_search` : ajout des articles sélectionnés aux entrées pour recommandation,
- `/save_from_search` : sauvegarde des articles sélectionnés dans le module SAVE,
- `/delete_from_search` : suppression des articles sélectionnés dans la liste du module SEARCH.

RECOMMEND

RECOMMEND est l'unique module de la phase de recommandation. Il a pour but de permettre la recommandation proprement dite en gérant la liste des références en entrée et en choissant la méthode de recommandation. Les deux méthodes proposées sont la similarité TF-IDF et les mots-clés similaires.

La méthode par similarité TF-IDF (*TF-IDF similarity*) utilise une requête de type `more_like_this` avec les références en entrée. Afin de ne pas avoir une requête trop lourde, le nombre de termes est limité à 100. De plus, pour limiter l'impact des *stop words*, le nombre maximale de documents pour un mot est limité à 1 000 000. Enfin, un légère priorité est accordée au titre. La requête utilisée est présentée dans le listing 6.4.

Listing 6.4 – Requête de recommandation par similarité TF-IDF.

```
{
 'size': 100,
 'query': {
```

```
  'more_like_this': {
   'fields': ['title^2', 'abstract'],
   'like': target_docs,
   'max_query_terms': 100,
   'max_doc_freq': 1000000
  }
 }
}
```

La méthodes par mots-clés similaires (*Common fields of study*) est basée sur le calcul de l'indice de Jaccard entre l'ensemble des mots-clés obtenu par union des ensembles de mots-clés des articles d'intérêt et les mots-clés des articles candidats. L'implémentation de la requête proprement dite est réalisée dans *Elasticsearch* via la combinaison d'une requête `function_score` et une requête `bool`. La requête complète est présentée dans le listing 6.5.

Listing 6.5 – Requête de recommandation par mots-clés communs.

```
{
 'size': size,
 'query': {
  'function_score': {
   'query': {
    'bool': {
     'should': should_queries,
     'minimum_should_match': 1
    }
   },
   'script_score': {
    'script': '1 / (' + str(nb_cit) + ' + doc["linked_papers"].size
       () - _score)'
   },
   'score_mode': 'multiply'
  }
 }
}
```

Routes associées :

- `/recommend` : lancement de la recommandation avec la méthode sélectionnée et les articles en entrée,
- `/remove_input` : suppression de la référénce de la liste des entrées pour recommandation.

EXPLOIT

EXPLOIT est le premier module de la phase d'exploitation des résultats. Sa fonction principale est la gestion des résultats de la recommandation. Outre la consultation, il permet notamment à l'utilisateur de modifier le classement selon ses objectifs de recherche bibliographique.

Le processus de classement comprend 4 étapes :

1. Les scores pour les critères de précision et de nouveauté sont calculés directement.
2. Les scores partiels pour les critères de diversité sont calculés en deux étapes. Le classement des recommandations est obtenu selon une approche gloutonne consistant à choisir à chaque itération l'article le moins similaire avec les articles déjà classés. Le score partiel pour chaque recommandation correspond à l'inverse de son rang dans le classement produit.
3. Les scores partiels sont normalisés ($score_{normalise} = (score - score_{min})/(score_{max} - score_{min})$) et combinés linéairement avec la pondération fournie par l'utilisateur pour produire les scores finaux.
4. Les recommandations sont classées selon les scores finaux.

Les critères de tri utilisés sont les suivants :

— La similarité textuelle avec le profil utilisateur (i.e. la sélection d'articles) : le score est basé sur la similarité cosinus entre les vecteurs TF-IDF obtenus à partir des titres et abstracts.
— La similarité topologique avec le profil utilisateur : le score est basé sur l'indice de Jaccard entre les ensembles de références et citations.
— La diversité de contenu infra-liste : la dissimilarité est basée sur le complément de la similarité cosinus entre les vecteurs TF-IDF à partir des titres et les abstracts.
— La diversité structurelle infra-liste : la dissimilarité est basée sur le complément de l'indice de Jaccard entre les ensembles de références et citations.
— La nouveauté temporelle : le score est calculé à partir de l'inverse de l'année de publication.
— La nouveauté via popularité inverse : le score est calculé à partir de la popularité inverse.

Routes associées :

— `/rank` : classement des articles sur base des poids fournis en paramètre,
— `/show` : accès au module EXPLOIT,
— `/add_from_exploit` : ajout des articles sélectionnés aux entrées pour recommandation,
— `/save_from_exploit` : sauvegarde des articles sélectionnés dans le module SAVE,
— `/delete_from_exploit` : suppression des articles sélectionnés dans la liste du module EXPLOIT.

DASHBOARD

DASHBOARD est le deuxième module de la phase d'exploitation des recommandations. Il permet d'accéder aux listes des auteurs, des organes de publication, des mots-clés et des années pour les articles recommandés, ordonnés par nombre d'occurrences.

Routes associées :

— `/dashboard` : accède au module DASHBOARD.

SAVE

SAVE est le troisième module de la phase d'exploitation des recommandations. Il permet de gérer la liste des résultats sauvegardés et notamment de les exporter au format BibTeX (via la librairie *Pybtex*).

Routes associées :

— `/save` : accès au module SAVE,
— `/export` : exportation des articles sélectionnés au format BibTeX,
— `/delete_from_save` : suppression des articles sélectionnés dans la liste du module SAVE.

6.3.4 Déploiement

Le prototype est déployé sur une machine virtuelle hébergée par le CETIC et accessible via l'adresse `https://stage-bibliographie.cetic.be/`. Il consiste en un serveur web *Nginx*, un serveur web WSGI *Gunicorn* exécutant l'application, une base de données PostgreSQL et une instance d'*Elasticsearch*. Le serveur *Nginx* est chargé de la redirection de port vers le serveur *Gunicorn* (`:80` vers `:5000`). Un diagramme de déploiement est présenté sur la figure 6.4.

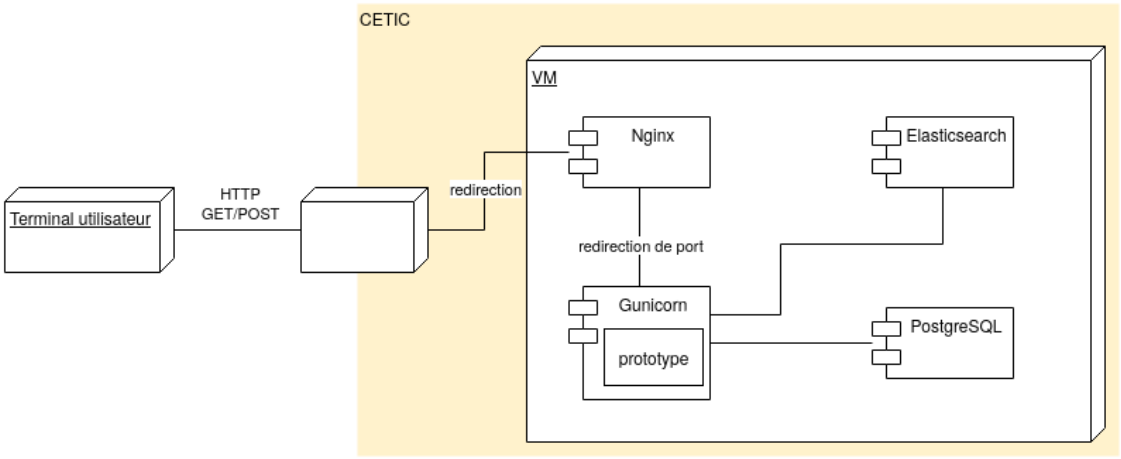

FIGURE 6.4 – Diagramme de déploiement du prototype.

Chapitre 7

Étude utilisateur

Ce chapitre est consacré à l'étude utilisateur, nécessaire pour évaluer l'intérêt du prototype. En effet, de nombreux aspects du prototype peuvent difficilement être évalués autrement que par l'intermédiaire de véritables utilisateurs. De plus, les limites de l'évaluation hors-ligne (évoquées dans la section 3.4) ne permettent pas conclure quant à la qualité réelle des recommandations.

Dans un premier temps, le processus de conception de l'étude utilisateur est présenté par l'intermédiaire de ces éléments constitutifs : la définition des objectifs, les modalités pratiques de l'étude et la construction du questionnaire. Et dans un second temps, les résultats sont présentés et discutés.

7.1 Conception de l'étude

7.1.1 Objectifs

Le premier objectif de l'étude utilisateur est d'évaluer l'intérêt général de l'outil développé pour les utilisateurs potentiels. Cet outil propose en effet une manière novatrice de faire de la recherche bibliographique qui doit être validée. Des résultats positifs sont évidemment une condition nécessaire à la poursuite du développement.

Le deuxième objectif concerne la qualité des recommandations produites. Deux méthodes sont proposées à l'utilisateur : la similarité TF-IDF et les mots-clés communs. Le choix de la première méthode est bien sûr le résultat de l'évaluation hors-ligne et il s'agit donc de vérifier si les recommandations proposées satisfont effectivement l'utilisateur. Le choix de proposer les mots-clés communs comme seconde méthode est motivé par le souhait de valider la mesure utilisée pour l'évaluation hors-ligne de la précision. L'idée est de montrer que cette mesure est un bon moyen d'évaluer la précision des recommandations, et par extension, de valider cet aspect du protocole d'évaluation hors-ligne.

Enfin, le troisième objectif concerne les fonctionnalités de tri. Il s'agit ici de vérifier si celles-ci améliorent effectivement le classement des recommandations selon les différents aspects correspondants. L'évaluation des fonctions de tri sera bien sûr influencée par le choix de la méthode. Celui-ci sera donc pris en compte dans l'analyse des résultats (dans la mesure du possible, i.e. si suffisamment de résultats).

7.1.2 Modalités pratiques

Concrètement, les participants doivent interagir avec le système de recommandation accessible en ligne [1] selon un scénario précis. Et ils remplissent ensuite un questionnaire d'évaluation (réalisé avec *Google Forms*).

Durée de l'étude

L'étude utilisateur a été mise à disposition sur la page d'accueil du prototype du vendredi 17 juillet au dimanche 16 août (4 semaines).

Participants sollicités

— Le personnel du CETIC (contacté par mail),
— Le personnel académique de la Faculté d'informatique de l'UNamur (contacté par mail),
— Les étudiants du bloc 2 du master 120 en sciences informatiques de l'UNamur (contactés via Facebook).

Scénario d'utilisation à évaluer

Afin de guider l'utilisation de l'outil par les participants à l'étude, un scénario sous forme d'une liste d'instructions est proposé. L'objectif est que les participants puissent expérimenter les différents aspects et fonctionnalités évalués. Lors des phases de recommandation et d'exploitation, le focus est mis sur les 10 premiers résultats parce qu'il s'agit du comportement de consultation le plus fréquemment décrit lors de l'analyse.

Instructions

1. Sélectionnez 1-5 articles représentatifs d'un sujet de recherche que vous souhaitez explorer via les modules SEARCH et/ou LOAD.
2. Choisissez une méthode de recommandation et lancez le processus via le module RECOMMEND.
3. Prenez connaissance des 10 premiers résultats affichés dans le module EXPLOIT.
4. Utilisez les fonctionnalités de tri pour modifier les 10 premiers résultats affichés.
5. Consultez le DASHBOARD.
6. Remplissez le questionnaire d'évaluation.
7. (Facultatif) Recommencez le processus à partir de l'étape 2 avec une autre méthode de recommandation.

7.1.3 Construction du questionnaire

La principale source d'inspiration pour la construction du questionnaire est l'étude utilisateur proposée par Sesagiri Raamkumar et al. [2017].

1. `https://stage-bibliographie.cetic.be/`

Échelle de valeurs utilisée pour les affirmations (échelle de Likert)

Le questionnaire est principalement constitué d'affirmations relatives aux fonctionnalités et aspects évalués, et par rapport auxquelles les participants sont invités à se positionner selon une échelle de Likert. Il s'agit d'une technique fréquemment utilisée dans la littérature [Sesagiri Raamkumar et al., 2017] et qui a l'avantage de pouvoir facilement quantifier les réponses obtenues. Quelques questions à choix multiples sont aussi utilisées pour la définition du profil. Enfin, les participants sont invités à fournir des commentaires libres sur l'outil en général, les méthodes de recommandation et les fonctionnalités de tri. Celles-ci ont pour but de fournir des idées d'amélioration pour un éventuel développement ultérieur.

Échelle de Likert utilisée :
- 1 : Totalement en désaccord,
- 2 : Plutôt en désaccord,
- 3 : Neutre,
- 4 : Plutôt d'accord,
- 5 : Totalement d'accord.

Profil utilisateur

Cette section a simplement pour but de catégoriser les participants afin de pouvoir distinguer les éventuelles variations dans les résultats en fonction du profil de l'utilisateur.

Description du profil utilisateur :
- Dans quelle structure travaillez-vous ? (choix entre : UNamur, CETIC, autre institution académique, autre centre/département de recherche, autre)
- Quel est votre niveau d'étude ? (choix entre : bachelier, master, phd, autre)
- Combien d'années d'expérience avez-vous en recherche bibliographique ? (choix entre : moins de 5 ans, 5-10 ans, plus de 10 ans)

Évaluation générale de l'outil

Cette section est consacrée à l'évaluation de l'outil en général. Elle distingue notamment la stratégie de recherche bibliographique proposée de l'outil proprement dit. L'idée étant de pouvoir évaluer l'intérêt de cette approche indépendamment de la réception du prototype dans sa version actuelle. Également dans une perspective de développement ultérieur, elle évalue aussi l'intérêt suscité par le dashboard.

Affirmations concernant l'outil en général :
- Cette manière de faire de la recherche bibliographique est une alternative intéressante à la recherche classique.
- Cet outil dans sa version actuelle est une alternative convaincante à un moteur de recherche classique.
- Vous appréciez le workflow général de l'outil (i.e. 1. sélection d'inputs, 2. recommandation, 3. exploitation des résultats).
- Vous appréciez la présentation des résultats.

— Les informations du dashboard vous semblent utiles en complément des articles recommandés.
— Avez-vous des commentaires à propose de l'outil en général ? (réponse libre)

Évaluation des méthodes de recommandation

Cette section a pour but d'évaluer la qualité des 10 premières recommandations selon différents aspects : la précision, la diversité, la présence d'articles de référence, la nouveauté temporelle et pour l'utilisateur, et la sérendipité.

Affirmations sur les 10 premiers résultats obtenus par la méthode de recommandation choisie :
— Quelle méthode de recommandation avez-vous utilisée ? (choix entre : similarité TF-IDF, mots-clés communs)
— La liste contient des articles sémantiquement proches du sujet de recherche.
— La liste contient des articles avec des approches variées du sujet d'intérêt.
— La liste contient des articles de référence sur le sujet de recherche.
— La liste contient des articles récents.
— La liste contient des articles que vous ne connaissez pas.
— La liste contient des articles que vous ne vous attendiez pas à voir.
— De manière générale, vous êtes satisfait de la liste proposée.
— Avez-vous des commentaires à propose des méthodes de recommandation ? (réponse libre)

Évaluation des fonctionnalités de tri

Cette section a pour but d'évaluer la capacité des fonctionnalités à améliorer les 10 premières recommandations. Chaque critère est évalué selon la propriété qu'il est sensé améliorer. Enfin, l'intérêt de combiner les critères de tri est aussi pris en compte. Cette fonctionnalité n'est cependant pas approfondie car cela aurait complexifié cette section, avec le risque d'avoir des réponses de moindre qualité.

Affirmations sur l'impact des possibilités de tri sur l'amélioration des 10 premiers résultats :
— Le tri par similarité de contenu avec les articles en entrée améliore la similarité sémantique avec le sujet de recherche.
— Le tri par similarité de références et citations avec les articles en entrée améliore la similarité sémantique avec le sujet de recherche.
— Le tri par diversité de contenu infra-liste améliore la diversité des approches du sujet d'intérêt.
— Le tri par diversité de références et citations infra-liste améliore la diversité des approches du sujet d'intérêt.
— Le tri par date de publication améliore la nouveauté des articles proposée (dans le sens d'inconnu pour vous).
— Le tri par popularité inverse améliore la nouveauté des articles proposée (dans le sens d'inconnu pour vous).
— Le tri par popularité inverse vous a permis d'améliorer, via une valeur négative, l'autorité des résultats (i.e. la présence d'articles de référence).
— La possibilité de combiner les critères de tri est un plus pour améliorer les 10 premiers résultats.

— Des fonctionnalités de filtrage des résultats seraient utiles.
— Avez-vous des commentaires à propos des fonctionnalités de tri ? (réponse libre)

7.2 Analyse des résultats

Malheureusement, l'étude utilisateur n'a pas rencontré un franc succès avec seulement 4 participants. Bien que 4 semaines semblent être une durée suffisante dans l'absolu, la période choisie et le contexte actuel expliquent sans doute ce faible nombre. Il est donc très difficile de pouvoir généraliser les résultats obtenus et d'en tirer des conclusions. Néanmoins, les réponses obtenues offrent quelques indications utiles pour l'éventuelle poursuite du développement de l'outil. Le détail des résultats de l'étude utilisateur est présent dans l'annexe A.4.

Profil des participants

3 participants appartiennent à la Faculté d'informatique de l'UNamur et 1 participant appartient à une autre institution académique. 2 participants ont un diplôme de master et 2 participants sont titulaires d'un doctorat. Enfin, 3 participants ont une expérience en recherche bibliographique inférieure à 5 ans et 1 participant a une expérience de plus de 10 ans.

Évaluation générale de l'outil

Les résultats obtenus (voir table 7.1) indiquent un intérêt pour cette stratégie de recherche bibliographique. Par contre, l'outil dans sa version actuelle n'est pas considéré comme suffisamment convaincant. Ceci s'explique notamment par l'utilisation d'un corpus d'articles fermés qui limite le champ des recommandations, l'absence de fonctionnalités de base pour un outil de ce type ou encore l'IHM qui n'a pas fait l'objet d'une recherche spécifique. La présentation des résultats suscite d'ailleurs des réactions plutôt contrastées. Néanmoins, l'appréciation générale du workflow de l'outil semble indiquer que celui-ci va dans la bonne direction quant à l'application de cette stratégie de recherche. Enfin, il est intéressant de constater que le dashboard suscite pas mal d'intérêt malgré qu'il soit encore très rudimentaire. Améliorer cette vue analytique des résultats pourrait donc être une piste de développement intéressante. D'autant plus que cet aspect est peu développé de manière générale dans les outils de recherche bibliographique, à l'exception de quelques uns comme *Dimensions*[2].

Évaluation des méthodes de recommandation

La méthode choisie par les 4 participants est la similarité TF-IDF. De manière générale, les résultats obtenus (voir table 7.2) indiquent que les participants sont plutôt satisfaits de la liste des 10 premières recommandations. Ce qui est également le cas du point de vue de la similarité sémantique, de la diversité, de la nouveauté (articles récents) et de la sérendipité. Par contre, les avis sont plus contrastés en ce qui concerne la présence d'articles de référence et d'articles que les participants ne connaissent pas.

À noter également que certains participants auraient souhaités davantage de feedback sur les recommandations, et notamment des informations sur la qualité des résultats. Ce qui confirme l'importance de l'explicabilité des recommandations déjà soulevée lors de la phase d'analyse. Cet aspect devra donc être privilégié par la suite.

2. `https://www.dimensions.ai/`

TABLE 7.1 – Affirmations concernant l'outil en général.

Affirmation	Moyenne	Écart moyen
Cette manière de faire de la recherche bibliographique est une alternative intéressante à la recherche classique.	3,5	0,5
Cet outil dans sa version actuelle est une alternative convaincante à un moteur de recherche classique.	2,5	0,5
Vous appréciez le workflow général de l'outil (i.e. 1. sélection d'inputs, 2. recommandation, 3. exploitation des résultats).	3,75	0,375
Vous appréciez la présentation des résultats.	3	1
Les informations du dashboard vous semblent utiles en complément des articles recommandés.	3,75	0,375

TABLE 7.2 – Affirmations sur les 10 premiers résultats obtenus par la méthode de recommandation choisie.

Affirmation	Moyenne	Écart moyen
La liste contient des articles sémantiquement proches du sujet de recherche.	3,75	0,75
La liste contient des articles avec des approches variées du sujet d'intérêt.	3,75	0,375
La liste contient des articles de référence sur le sujet de recherche.	3,25	0,375
La liste contient des articles récents.	4	0
La liste contient des articles que vous ne connaissez pas.	3,25	0,75
La liste contient des articles que vous ne vous attendiez pas à voir.	3,75	0,375
De manière générale, vous êtes satisfait de la liste proposée.	3,5	0,5

TABLE 7.3 – Affirmations sur l'impact des possibilités de tri sur l'amélioration des 10 premiers résultats.

Affirmation	Moyenne	Écart-type
Le tri par similarité de contenu avec les articles en entrée améliore la similarité sémantique avec le sujet de recherche.	2,75	0,375
Le tri par similarité de références et citations avec les articles en entrée améliore la similarité sémantique avec le sujet de recherche.	3	0
Le tri par diversité de contenu infra-liste améliore la diversité des approches du sujet d'intérêt.	2,75	0,375
Le tri par diversité de références et citations infra-liste améliore la diversité des approches du sujet d'intérêt.	2,5	0,5
Le tri par date de publication améliore la nouveauté des articles proposée (dans le sens d'inconnu pour vous).	3,75	0,375
Le tri par popularité inverse améliore la nouveauté des articles proposée (dans le sens d'inconnu pour vous).	3,5	0,5
Le tri par popularité inverse vous a permis d'améliorer, via une valeur négative, l'autorité des résultats (i.e. la présence d'articles de référence).	3	0
La possibilité de combiner les critères de tri est un plus pour améliorer les 10 premiers résultats.	3	0,5
Des fonctionnalités de filtrage des résultats seraient utiles.	3,5	0,5

Évaluation des fonctionnalités de tri

Les résultats obtenus (voir table 7.3) sont davantage négatifs en ce qui concerne les fonctionnalités de tri. Le reclassement via les critères de précision et de diversité ne donne pas vraiment satisfaction. En ce qui concerne la précision par similarité textuelle, l'explication est sans doute que la méthode de recommandation est basée également sur la similarité textuelle et qu'il est donc difficile d'améliorer le classement initial. Pour la précision par similarité topologique, une explication possible est la moindre efficacité de ce critère pour améliorer la similarité sémantique de manière générale [Bhattacharya et al., 2020]. Par contre, les résultats négatifs concernant la diversité sont plus étonnants et demanderaient d'être investigués. Enfin, les critères de nouveauté ont davantage satisfait les participants.

Mais l'enseignement le plus important provient sans doute des commentaires, qui confirment d'ailleurs les premiers retours informels sur le prototype. Il s'agit du caractère peu intuitif des fonctionnalités de tri telles que proposées dans le prototype. Les retours plutôt neutres pour la combinaison des critères pourraient d'ailleurs s'expliquer par la difficulté à en comprendre l'intérêt. Ces commentaires et le peu de satisfaction par rapport à l'efficacité des différents critères de tri poussent à questionner la stratégie employée. Il s'agirait de voir si une simple refonte des fonctionnalités de tri, notamment en simplifiant leur utilisation, susciterait davantage d'adhésion. Ou si la stratégie choisie, c'est-à-dire la recommandation via une méthode unique et le tri des résultats selon les besoins de l'utilisateur, est la plus adéquate.

Chapitre 8

Conclusion et perspectives

Cette conclusion est l'occasion de faire le bilan du travail réalisé et d'évoquer quelques perspectives d'amélioration. Tout d'abord, les principaux résultats de la comparaison des méthodes sont rappelés ainsi que les choix d'interprétation opérés. Ensuite, le protocole d'évaluation lui-même est évoqué en mettant en avant son intérêt par rapport à d'autres approches et ses limites. Enfin, un bilan du prototype est réalisé à partir des résultats de l'étude utilisateur et des pistes d'améliorations sont suggérées.

Comparaison des méthodes

Le principal enseignement de cette comparaison est que les méthodes *state-of-art* ne sont pas forcément meilleures que les méthodes plus simples dans les différents aspects évalués. Ce constat doit être néanmoins relativisé. En effet, l'objectif était de réaliser une évaluation à large spectre en comparant un large éventail de méthodes appartenant à des classes de recommandation différentes. Il était donc impossible d'optimiser le choix des paramètres et donc de comparer les variantes les plus efficaces. Certaines méthodes mériteraient donc davantage d'investigation avant d'être écartées.

Le filtrage hybride, qui est la classe de recommandation la plus riche et la plus active dans le domaine (voir section 2.2.7), n'a été explorée que très partiellement. Il s'agit donc sans doute de la piste à privilégier afin d'identifier de nouvelles méthodes prometteuses. Même si ces dernières sont en général plutôt complexes à mettre en œuvre. Par ailleurs, ce travail s'est concentré sur les méthodes de recommandation dans le domaine de la littérature scientifique. L'exploration d'autres domaines où des systèmes de recommandation sont développés pourrait donc être également une piste intéressante.

La principale difficulté dans la sélection des méthodes pour le prototype a été la manière de considérer les multiples aspects et mesures envisagés. Le choix a été de privilégier les résultats de la précision de contenu étant donné que cet aspect est particulièrement important pour les utilisateurs potentiels comme le montre l'analyse (voir chapitre 4). D'autres approches intégrant des aspects comme la nouveauté auraient pu être envisagées.

Un autre choix posé est d'évaluer la qualité globale des recommandations plutôt que de prendre en compte le classement comme c'est généralement le cas dans la littérature (voir section 3.4). Il est motivé par la stratégie de recherche employée qui combine la recommandation et le reclassement selon les besoins de l'utilisateur. Le classement initial des items (i.e. après recommandation) est donc moins important dans ce cas. Par contre, la qualité globale est essentielle afin de garantir la présence d'items pertinents indépendamment des critères de tri utilisés. Ce choix devrait cependant être

reconsidéré si une stratégie s'appuyant davantage sur la recommandation pour répondre aux besoins des utilisateurs était employée. Une mesure de classement comme le gain cumulatif discontinu pourrait être préférée dans ce cas. D'autant plus que cette dernière peut intégrer différents aspects par le biais de la fonction d'utilité choisie.

Protocole d'évaluation hors-ligne

L'intérêt majeur du protocole d'évaluation hors-ligne proposé est de permettre la comparaison de méthodes appartenant à des classes différentes. Ce protocole est basé sur l'estimation de la qualité des recommandations par le biais de mesures ne nécessitant pas d'être comparées à des listes de référence. Ce qui permet de contourner les différents biais introduits par les mesures d'évaluation supervisées (voir section 3.4). Ce protocole est donc clairement une manière de répondre à la problématique de l'absence de consensus sur les méthodes les plus performantes.

La plupart des mesures employées sont reconnues dans la littérature. Ce n'est cependant pas le cas de l'utilisation des mots-clés pour estimer la précision de contenu (bien qu'il existe quelques exemples comme le framework CITREC utilisant l'indexation MeSH [Gipp and Meuschke, 2015]). Cette mesure devrait donc être l'objet d'une validation plus systématique. Il serait également intéressant de voir si cette mesure peut être généralisée à d'autres types d'indexation comme la classification ACM. Ce qui permettrait de diversifier les jeux de données potentiellement utilisables et donc d'améliorer la robustesse des comparaisons.

Concernant les visualisations utilisées pour la présentation des résultats, les jeux de données ont été distingués afin de montrer leur impact sur la variabilité des classements. Ce choix rend cependant la lecture des résultats moins claire et l'interprétation plus complexe. Il pourrait donc être remis en cause, notamment si davantage de jeux de données sont utilisés.

Enfin, le protocole pourrait faire l'objet d'un développement spécifique afin de le rendre plus robuste, plus facilement utilisable et paramétrable afin de répondre aux différents cas d'utilisation. Et il pourrait dès lors être mis à la disposition de la communauté par le biais d'une plateforme comme GitHub par exemple.

Bilan du prototype et perspectives de développement

Le bilan du prototype est évidemment fort subjectif étant donné le faible de nombre de participants à l'étude utilisateur.

Le premier point positif est la stratégie de recherche proposée qui recueille un certain assentiment de la part des participants. Le prototype doit cependant encore être amélioré. Les deux principaux axes pointés sont un corpus d'articles beaucoup plus important et avec des possibilités de mise-à-jour, et l'amélioration substantielle de l'expérience utilisateur. D'autres fonctionnalités classiques pour ce genre d'outil sont également absentes (car non essentielles dans un premier temps) et devraient être ajoutées. Par exemple, la possibilité de consulter les informations relatives aux différentes entités (articles, personnes, organes, etc.) et la création de comptes utilisateurs permettant de gérer ses recherches.

Plus spécifiquement, le dashboard a suscité par mal d'intérêt bien qu'il soit assez rudimentaire. Il avait notamment pour fonction d'identifier les personnes et les organes importants, de montrer l'activité du domaine dans le temps, et d'identifier la terminologie employée. Cet intérêt est peut-être le signe de l'importance des besoins davantage analytiques en recherche bibliographique pour les utilisateurs potentiels, comme par exemple la localisation de la recherche ou l'évaluation de

la richesse pour un domaine spécifique. Outre l'amélioration des fonctionnalités déjà présentes, il pourrait donc être opportun de développer davantage le dashboard.

Le second point positif est le retour des participants sur les recommandations qui semble indiquer une certaine efficacité de la méthode choisie. Ces derniers semblent satisfaits de la précision, de la diversité, de la nouveauté et de la sérendipité des recommandations proposées. Ce qui est une bonne indication de la capacité de cette méthode à permettre la compréhension générale du domaine et la découverte des challenges et des opportunités. Par contre, cette méthode semble avoir plus de difficulté à identifier les articles de référence.

Par contre, le principal point négatif est que les fonctionnalités de tri ne suscitent guère d'enthousiasme de la part des participants (à l'exception des critères de nouveauté). Il est cependant difficile à ce stade de savoir si c'est le fonctionnement même du tri qui est en cause, ou la manière dont celui-ci est présenté (i.e. l'interface). Ou si c'est la stratégie de recherche même qui devrait être adaptée. Elle pourrait par exemple permettre de paramétrer la recommandation en fonction des besoins et simplifier les possibilités de tri en phase d'exploitation des résultats.

Bibliographie

Adomavicius, G. and Kwon, Y. [2015], Multi-Criteria Recommender Systems, *in* 'Recommender Systems Handbook', Springer US, Boston, MA, pp. 847–880.

Aggarwal, C. C. [2016], *Recommender Systems : The Textbook*, 1st edn, Springer International Publishing.

Alfarhood, M. and Cheng, J. [2019], Collaborative Attentive Autoencoder for Scientific Article Recommendation, *in* '18th IEEE International Conference on Machine Learning and Applications (ICMLA)', Boca Raton, Florida, USA.

Alshareef, A. M., Alhamid, M. F. and El Saddik, A. [2019], 'Toward citation recommender systems considering the article impact in the extended nearby citation network', *Peer-to-Peer Networking and Applications* **12**(5), 1336–1345.

Alvarez, J. E. and Bast, H. [2017], A review of word embedding and document similarity algorithms applied to academic text, PhD thesis, Univrsity of Freiburg.

Amami, M., Faiz, R., Stella, F. and Pasi, G. [2017], 'A graph based approach to scientific paper recommendation', *Proceedings - 2017 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2017* pp. 777–782.

Amami, M., Pasi, G., Stella, F. and Faiz, R. [2016], An LDA-Based Approach to Scientific Paper Recommendation, Vol. 9612 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, pp. 200–210.

Amsler, R. A. [1972], *Applications of citation-based automatic classification*, Linguistics Research Center, University of Texas at Austin.

Ayala-Gómez, F., Daróczy, B., Benczúr, A., Mathioudakis, M. and Gionis, A. [2018], 'Global citation recommendation using knowledge graphs', *Journal of Intelligent & Fuzzy Systems* **34**(5), 3089–3100.

Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X. and Xia, F. [2019], 'Scientific Paper Recommendation : A Survey', *IEEE Access* **7**, 9324–9339.

Beel, J., Carevic, Z., Schaible, J. and Neusch, G. [2017], 'RARD : The related-article recommendation dataset', *D-Lib Magazine* **23**(7-8).

Beel, J., Collins, A., Kopp, O., Dietz, L. W. and Knoth, P. [2019], 'Online evaluations for everyone : Mr. Dlib's living lab for scholarly recommendations', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **11438 LNCS**(13), 213–219.

Beel, J., Dinesh, S., Mayr, P., Carevic, Z. and Raghvendra, J. [2017], 'Stereotype and Most-Popular Recommendations in the Digital Library Sowiport', *Proceedings of the 15th International Symposium of Information Science (ISI 2017)* (March 2017), 96–108.

Beel, J., Gipp, B., Langer, S. and Breitinger, C. [2016], 'Research-paper recommender systems : a literature survey', *International Journal on Digital Libraries* **17**(4), 305–338.

Beel, J. and Langer, S. [2015], A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems, *in* 'Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)', Vol. 9316, pp. 153–168.

Beel, J., Smyth, B. and Collins, A. [2019], 'Rard II : The 94 million related-article recommendation dataset', *CEUR Workshop Proceedings* **2360**(13).

Bellogín, A. and Said, A. [2017], *Recommender Systems Evaluation*, Springer New York, New York, NY, pp. 1–18.

Bhagavatula, C., Feldman, S., Power, R. and Ammar, W. [2018], 'Content-Based Citation Recommendation', *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)* pp. 238–251.

Bhattacharya, P., Ghosh, K., Pal, A. and Ghosh, S. [2020], 'Methods for Computing Legal Document Similarity : A Comparative Study', *Preprint* (i).

Blei, D. M., Ng, A. Y. and Jordan, M. I. [2003], 'Latent dirichlet allocation', *Journal of machine Learning research* **3**(Jan), 993–1022.

Bobadilla, J., Ortega, F., Hernando, A. and Gutiérrez, A. [2013], 'Recommender systems survey', *Knowledge-Based Systems* **46**, 109–132.

Bornmann, L. and Mutz, R. [2015], 'Growth rates of modern science : A bibliometric analysis based on the number of publications and cited references', *Journal of the Association for Information Science and Technology* **66**(11), 2215–2222.

Boyack, K. W. and Klavans, R. [2010], 'Co-citation analysis, bibliographic coupling, and direct citation : Which citation approach represents the research front most accurately ?', *Journal of the American Society for Information Science and Technology* **61**(12), 2389–2404.

Bridge, D., Göker, M. H., McGinty, L. and Smyth, B. [2005], 'Case-based recommender systems', *The Knowledge Engineering Review* **20**(3), 315–320.

Brochier, R. [2019], 'Representation Learning for Recommender Systems with Application to the Scientific Literature', pp. 12–16.

Bulut, B., Kaya, B., Alhajj, R. and Kaya, M. [2018], A Paper Recommendation System Based on User's Research Interests, *in* '2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)', IEEE, pp. 911–915.

Burke, R. [2002], 'Hybrid Recommender Systems : Survey and Experiments', *User Modeling and User-Adapted Interaction* **12**(4), 331–370.

Cai, X., Han, J., Li, W., Zhang, R., Pan, S. and Yang, L. [2018], 'A Three-Layered Mutually Reinforced Model for Personalized Citation Recommendation', *IEEE Transactions on Neural Networks and Learning Systems* **29**(12), 6026–6037.

Cai, X., Han, J., Pan, S. and Yang, L. [2018], 'Heterogeneous Information Network Embedding based Personalized Query-Focused Astronomy Reference Paper Recommendation', *International Journal of Computational Intelligence Systems* **11**(1), 591.

Cai, X., Han, J. and Yang, L. [2018], Generative Adversarial Network based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation, *in* '[AAAI2018]Proceedings of the Thirtiy-second AAAI Conference on Artificial Intelligence', pp. 5747–5754.

Cai, X., Zheng, Y., Yang, L., Dai, T. and Guo, L. [2019], 'Bibliographic Network Representation Based Personalized Citation Recommendation', *IEEE Access* **7**, 457–467.

Castells, P., Hurley, N. J. and Vargas, S. [2015], Novelty and Diversity in Recommender Systems, *in* 'Recommender Systems Handbook', Springer US, Boston, MA, pp. 881–918.

Chakraborty, T., Krishna, A., Singh, M., Ganguly, N., Goyal, P. and Mukherjee, A. [2016], FeRoSA : A Faceted Recommendation System for Scientific Articles, *in* 'Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)', Vol. 9652 LNAI, pp. 528–541.

Chakraborty, T., Modani, N., Narayanam, R. and Nagar, S. [2015], DiSCern : A diversified citation recommendation system for scientific queries, *in* '2015 IEEE 31st International Conference on Data Engineering', Vol. 2015-May, IEEE, pp. 555–566.

Champiri, Z. D., Shahamiri, S. R. and Salim, S. S. B. [2015], 'A systematic review of scholar context-aware recommender systems', *Expert Systems with Applications* **42**(3), 1743–1758.

Chen, J. and Ban, Z. [2016], Literature recommendation by researchers' publication analysis, *in* '2016 IEEE International Conference on Information and Automation (ICIA)', number August, IEEE, pp. 1964–1969.

Chen, L. and Pu, P. [2012], 'Critiquing-based recommenders : survey and emerging trends', *User Modeling and User-Adapted Interaction* **22**(1-2), 125–150.

Chen, T. T. and Lee, M. [2018], Research Paper Recommender Systems on Big Scholarly Data, Vol. 7457 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 251–260.

Chen, X. [2010], 'The Declining Value of Subscription-based Abstracting and Indexing Services in the New Knowledge Dissemination Era', *Serials Review* **36**(2), 79–85.

Couto, T., Cristo, M., Gonçalves, M. A., Calado, P., Ziviani, N., Moura, E. and Ribeiro-Neto, B. [2006], 'A comparative study of citations and links in document classification', *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries* **2006**, 75–84.

Dai, T., Gao, T., Zhu, L., Cai, X. and Pan, S. [2018], 'Low-rank and sparse matrix factorization for scientific paper recommendation in heterogeneous network', *IEEE Access* **6**, 59015–59030.

Dai, T., Zhu, L., Cai, X., Pan, S. and Yuan, S. [2018], 'Explore semantic topics and author communities for citation recommendation in bipartite bibliographic network', *Journal of Ambient Intelligence and Humanized Computing* **9**(4), 957–975.

de Gemmis, M., Lops, P. and Polignano, M. [2017], Recommender Systems, Basics Of, *in* 'Encyclopedia of Social Network Analysis and Mining', Springer New York, New York, NY, pp. 1–13.

De Nart, D. and Tasso, C. [2014], 'A personalized concept-driven Recommender System for scientific libraries', *Procedia Computer Science* **38**(C), 84–91.

Demšar, J. [2006], 'Statistical comparisons of classifiers over multiple data sets', *Journal of Machine learning research* **7**(Jan), 1–30.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. [2018], 'Bert : Pre-training of deep bidirectional transformers for language understanding'.

Dhanda, M. and Verma, V. [2016], 'Recommender System for Academic Literature with Incremental Dataset', *Procedia Computer Science* **89**, 483–491.

Färber, M. and Sampath, A. [2020], 'HybridCite : A Hybrid Model for Context-Aware Citation Recommendation'.

Felfernig, A. and Burke, R. [2008], 'Constraint-based recommender systems : Technologies and research issues', *ACM International Conference Proceeding Series* .

Felfernig, A., Friedrich, G., Jannach, D. and Zanker, M. [2015], Constraint-Based Recommender Systems, *in* 'Recommender Systems Handbook', Springer US, Boston, MA, pp. 161–190.

Ganguly, S. and Pudi, V. [2017], 'Paper2vec : Combining graph and text information for scientific paper representation', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **10193 LNCS**, 383–395.

Gilchrist, A. [2003], 'Thesauri, taxonomies and ontologies – an etymological note', *Journal of Documentation* **59**(1), 7–18.

Gipp, B. and Meuschke, N. [2015], 'CITREC : An Evaluation Framework for Citation-Based Similarity Measures based on TREC Genomics and PubMed Central', *Proceedings of the iConference 2015* pp. 24–27.

Goyal, P. and Ferrara, E. [2018], 'Graph embedding techniques, applications, and performance : A survey', *Knowledge-Based Systems* **151**, 78–94.

Grover, A. and Leskovec, J. [2016], node2vec : Scalable Feature Learning for Networks, *in* 'Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16', Vol. 13-17-Augu, ACM Press, New York, New York, USA, pp. 855–864.

Gunawardana, A. and Shani, G. [2015], *Evaluating Recommender Systems*, Springer US, Boston, MA, pp. 265–308.

Gupta, S. and Varma, V. [2017], Scientific Article Recommendation by using Distributed Representations of Text and Graph, *in* 'Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion', ACM Press, New York, New York, USA, pp. 1267–1268.

Gusenbauer, M. [2019], 'Google Scholar to overshadow them all ? Comparing the sizes of 12 academic search engines and bibliographic databases', *Scientometrics* **118**(1), 177–214.

Hamilton, W. L., Ying, R. and Leskovec, J. [2017], 'Representation Learning on Graphs : Methods and Applications', pp. 1–24.

Haruna, K., Akmar Ismail, M., Damiasih, D., Sutopo, J. and Herawan, T. [2017], 'A collaborative approach for research paper recommender system', *PLOS ONE* **12**(10), e0184516.

Haveliwala, T. H. [2003], 'Topic-sensitive pagerank : A context-sensitive ranking algorithm for web search', *IEEE transactions on knowledge and data engineering* **15**(4), 784–796.

Huang, W., Wu, Z., Liang, C., Mitra, P. and Giles, C. L. [2015], A Neural Probabilistic Model for Context Based Citation Recommendation, *in* 'AAAI 2015 : Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence', pp. 2404–2410.

Isinkaye, F., Folajimi, Y. and Ojokoh, B. [2015], 'Recommendation systems : Principles, methods and evaluation', *Egyptian Informatics Journal* **16**(3), 261 – 273.

Jardine, J. and Teufel, S. [2014], Topical PageRank : A Model of Scientific Expertise for Bibliographic Search, *in* 'Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics', Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 501–510.

Jia, H. and Saule, E. [2018], 'Graph Embedding for Citation Recommendation', pp. 1–17.

Jiang, Z., Liu, X. and Gao, L. [2015], Chronological Citation Recommendation with Information-Need Shifting, *in* 'Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM '15', ACM Press, New York, New York, USA, pp. 1291–1300.

Jinha, A. [2010], 'Article 50 million : An estimate of the number of scholarly articles in existence', *Learned Publishing* **23**(3), 258–263.

Jones, K. S. [1972], 'A statistical interpretation of term specificity and its application in retrieval', *Journal of documentation* .

Kaminskas, M. and Bridge, D. [2016], 'Diversity, Serendipity, Novelty, and Coverage', *ACM Transactions on Interactive Intelligent Systems* **7**(1), 1–42.

Kanakia, A., Shen, Z., Eide, D. and Wang, K. [2019], A Scalable Hybrid Research Paper Recommender System for Microsoft Academic, *in* 'The World Wide Web Conference on - WWW '19', ACM Press, New York, New York, USA, pp. 2893–2899.

Kessler, M. M. [1963], 'Bibliographic coupling between scientific papers', *American Documentation* **14**(1), 10–25.

Khabsa, M. and Giles, C. L. [2014], 'The Number of Scholarly Documents on the Public Web', *PLoS ONE* **9**(5), e93949.

Khabsa, M., Wu, Z. and Giles, C. L. [2016], Towards Better Understanding of Academic Search, *in* 'Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries - JCDL '16', ACM Press, New York, New York, USA, pp. 111–114.

Khusro, S., Ali, Z. and Ullah, I. [2016], Recommender Systems : Issues, Challenges, and Research Opportunities, *in* 'Information Science and Applications (ICISA) 2016', pp. 1179–1189.

Knijnenburg, B. P. and Willemsen, M. C. [2015], Evaluating Recommender Systems with User Experiments, *in* 'Recommender Systems Handbook', Springer US, Boston, MA, pp. 309–352.

Kobayashi, Y., Shimbo, M. and Matsumoto, Y. [2018], Citation Recommendation Using Distributed Representation of Discourse Facets in Scientific Articles, *in* 'Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries - JCDL '18', ACM Press, New York, New York, USA, pp. 243–251.

Kong, X., Mao, M., Wang, W., Liu, J. and Xu, B. [2018], 'VOPRec : Vector Representation Learning of Papers with Text Information and Structural Identity for Recommendation', *IEEE Transactions on Emerging Topics in Computing* **6750**(c), 1–1.

Koren, Y., Bell, R. and Volinsky, C. [2009], 'Matrix Factorization Techniques for Recommender Systems', *Computer* **42**(8), 30–37.

Kotkov, D., Wang, S. and Veijalainen, J. [2016], 'A survey of serendipity in recommender systems', *Knowledge-Based Systems* **111**, 180–192.

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. and Brown, D. [2019], 'Text classification algorithms : A survey', *Information (Switzerland)* **10**(4), 1–68.

Kusner, M. J., Sun, Y., Kolkin, N. I. and Weinberger, K. Q. [2015], From word embeddings to document distances, *in* 'ICML'15 : Proceedings of the 32nd International Conference on International Conference on Machine Learning', Vol. 37, pp. 957–966.

Le, M., Kayal, S. and Douglas, A. [2019], 'The impact of recommenders on scientific article discovery : The case of Mendeley suggest', *CEUR Workshop Proceedings* **2462**.

Le, Q. and Mikolov, T. [2014], Distributed representations of sentences and documents, *in* E. P. Xing and T. Jebara, eds, 'Proceedings of the 31st International Conference on Machine Learning', Vol. 32 of *Proceedings of Machine Learning Research*, PMLR, Bejing, China, pp. 1188–1196.

Lee, J., Lee, K., Kim, J. G. and Kim, S. [2015], Personalized Academic Research Paper Recommendation System, *in* 'Proceedings of the 6th International Workshop on Social Recommender Systems'.

Ley, M. [2009], 'DBLP', *Proceedings of the VLDB Endowment* **2**(2), 1493–1500.

Liu, H., Kong, X., Bai, X., Wang, W., Bekele, T. M. and Xia, F. [2015], 'Context-Based Collaborative Filtering for Citation Recommendation', *IEEE Access* **3**, 1695–1703.

Liu, H., Yang, Z., Lee, I., Xu, Z., Yu, S. and Xia, F. [2015], CAR : Incorporating Filtered Citation Relations for Scientific Article Recommendation, *in* '2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)', IEEE, pp. 513–518.

Liu, X., Yu, Y., Guo, C. and Sun, Y. [2014], Meta-Path-Based Ranking with Pseudo Relevance Feedback on Heterogeneous Graph for Citation Recommendation, *in* 'Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14', ACM Press, pp. 121–130.

Lo, K., Wang, L. L., Neumann, M., Kinney, R. and Weld, D. S. [2019], 'S2ORC : The Semantic Scholar Open Research Corpus', pp. 4969–4983.

Lu, J., Wu, D., Mao, M., Wang, W. and Zhang, G. [2015], 'Recommender system application developments : A survey', *Decision Support Systems* **74**, 12–32.

Ma, X. and Wang, R. [2019], 'Personalized Scientific Paper Recommendation Based on Heterogeneous Graph Representation', *IEEE Access* **7**, 79887–79894.

Melville, P. and Sindhwani, V. [2017], Recommender Systems, *in* K. J. Kim and N. Joukov, eds, 'Encyclopedia of Machine Learning and Data Mining', Vol. 376 of *Lecture Notes in Electrical Engineering*, Springer US, Boston, MA, pp. 1056–1066.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. [2013], 'Efficient estimation of word representations in vector space', *arXiv preprint arXiv :1301.3781* .

Mohamed Hassan, H. A., Sansonetti, G., Gasparetti, F., Micarelli, A. and Beel, J. [2019], BERT, ELMo, use and infersent sentence encoders : The Panacea for research-paper recommendation ?, *in* '13th ACM Conference on Recommender Systems', Vol. 2431, Copenhagen, Denmark, pp. 6–10.

Mu, D., Guo, L., Cai, X. and Hao, F. [2018], 'Query-Focused Personalized Citation Recommendation With Mutually Reinforced Ranking', *IEEE Access* **6**, 3107–3119.

Neethukrishnan, K. V. and Swaraj, K. P. [2017], 'Ontology based research paper recommendation using personal ontology similarity method', *Proceedings of the 2017 2nd IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2017* pp. 1–4.

Nogueira, R., Jiang, Z., Cho, K. and Lin, J. [2020], 'Navigation-Based Candidate Expansion and Pretrained Language Models for Citation Recommendation', pp. 1–12.

Ortega, F., Bobadilla, J., Gutierrez, A., Hurtado, R. and Li, X. [2018], 'Artificial Intelligence Scientific Documentation Dataset for Recommender Systems', *IEEE Access* **6**(c), 48543–48555.

Page, L., Brin, S., Motwani, R. and Winograd, T. [1999], The pagerank citation ranking : Bringing order to the web., Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Pan, L., Dai, X., Huang, S. and Chen, J. [2015], Academic Paper Recommendation Based on Heterogeneous Graph, *in* 'Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)', Vol. 9427, pp. 381–392.

Park, D. H., Kim, H. K., Choi, I. Y. and Kim, J. K. [2012], 'A literature review and classification of recommender systems research', *Expert Systems with Applications* **39**(11), 10059–10072.

Perozzi, B., Al-Rfou, R. and Skiena, S. [2014], Deepwalk : Online learning of social representations, *in* 'Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '14, ACM, New York, NY, USA, pp. 701–710.

Pu, P., Chen, L. and Hu, R. [2011], A user-centric evaluation framework for recommender systems, *in* 'Proceedings of the fifth ACM conference on Recommender systems - RecSys '11', ACM Press, New York, New York, USA, p. 157.

Radev, D. R., Muthukrishnan, P., Qazvinian, V. and Abu-Jbara, A. [2013], 'The ACL anthology network corpus', *Language Resources and Evaluation* **47**(4), 919–944.

Ravi, K. M., Mori, J. and Sakata, I. [2017], Cross-Domain Academic Paper Recommendation by Semantic Linkage Approach Using Text Analysis and Recurrent Neural Networks, *in* '2017 Portland International Conference on Management of Engineering and Technology (PICMET)', Vol. 2017-Janua, IEEE, pp. 1–10.

Ribeiro, M. T., Ziviani, N., Moura, E. S. D., Hata, I., Lacerda, A. and Veloso, A. [2014], 'Multiobjective Pareto-Efficient Approaches for Recommender Systems', *ACM Transactions on Intelligent Systems and Technology* **5**(4), 1–20.

Ricci, F. [2017], Recommender Systems : Models and Techniques, *in* 'Encyclopedia of Social Network Analysis and Mining', Springer New York, New York, NY, pp. 1–12.

Ricci, F., Shapira, B. and Rokach, L. [2015], Recommender Systems : Introduction and Challenges, *in* 'Recommender Systems Handbook, Second Edition', chapter 1, pp. 1–34.

Robertson, S. and Zaragoza, H. [2010], 'The Probabilistic Relevance Framework : BM25 and Beyond', *Foundations and Trends® in Information Retrieval* **3**(4), 333–389.

Rodriguez, M., Posse, C. and Zhang, E. [2012], Multiple objective optimization in recommender systems, *in* 'Proceedings of the sixth ACM conference on Recommender systems - RecSys '12', ACM Press, New York, New York, USA, p. 11.

Sesagiri Raamkumar, A. and Foo, S. [2018], Multi-method Evaluation in Scientific Paper Recommender Systems, *in* 'Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization', ACM, New York, NY, USA, pp. 179–182.

Sesagiri Raamkumar, A., Foo, S. and Pang, N. [2017], 'Using author-specified keywords in building an initial reading list of research papers in scientific paper retrieval and recommender systems', *Information Processing & Management* **53**(3), 577–594.

Shahmirzadi, O., Lugowski, A. and Younge, K. [2019], Text Similarity in Vector Space Models : A Comparative Study, *in* '2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)', IEEE, pp. 659–666.

Shen, Z., Ma, H. and Wang, K. [2018], A Web-scale system for scientific knowledge exploration, *in* 'Proceedings of ACL 2018, System Demonstrations', Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 87–92.

Small, H. [1973], 'Co-citation in the scientific literature : A new measure of the relationship between two documents', *Journal of the American Society for Information Science* **24**(4), 265–269.

Son, J. and Kim, S. B. [2018], 'Academic paper recommender system using multilevel simultaneous citation networks', *Decision Support Systems* **105**, 24–33.

Steinert, L. and Hoppe, H. U. [2016], 'A comparative analysis of network-based similarity measures for scientific paper recommendations', *Proceedings - 2016 3rd European Network Intelligence Conference, ENIC 2016* pp. 17–24.

Sternitzke, C. and Bergmann, I. [2009], 'Similarity measures for document mapping : A comparative study on the level of an individual scientist', *Scientometrics* **78**(1), 113–130.

Sugiyama, K. and Kan, M. Y. [2015], 'A comprehensive evaluation of scholarly paper recommendation using potential citation papers', *International Journal on Digital Libraries* **16**(2), 91–109.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L. and Su, Z. [2008], ArnetMiner : Extraction and Mining of Academic Social Networks, *in* 'Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08', ACM Press, New York, New York, USA, p. 990.

Tian, H. and Zhuo, H. H. [2017], 'Paper2vec : Citation-Context Based Document Distributed Representation for Scholar Recommendation'.

Wang, C., Wang, D., Feng, S., Zhang, Y. and Liu, H. [2017], A novel approach for paper recommendation based on rough-fuzzy set theory, *in* '2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)', Vol. 1, IEEE, pp. 1435–1442.

Wang, D., Liang, Y., Xu, D., Feng, X. and Guan, R. [2018], 'A content-based recommender system for computer science publications', *Knowledge-Based Systems* **157**(February 2017), 1–9.

Wang, H., Chen, B. and Li, W.-J. [2013], Collaborative topic regression with social regularization for tag recommendation, *in* 'IJCAI'.

West, J. D., Wesley-Smith, I. and Bergstrom, C. T. [2016], 'A Recommendation System Based on Hierarchical Clustering of an Article-Level Citation Network', *IEEE Transactions on Big Data* **2**(2), 113–123.

Wu, J., Liang, C., Yang, H. and Giles, C. L. [2015], 'CiteSeerX Data : Semanticizing Scholarly Papers'.

Xia, F., Liu, H., Lee, I. and Cao, L. [2016], 'Scientific Article Recommendation : Exploiting Common Author Relations and Historical Preferences', *IEEE Transactions on Big Data* **2**(2), 101–112.

Yang, L., Zhang, Z., Cai, X. and Guo, L. [2019], 'Citation Recommendation as Edge Prediction in Heterogeneous Bibliographic Network : A Network Representation Approach', *IEEE Access* **7**, 23232–23239.

Yang, L., Zheng, Y., Cai, X., Dai, H., Mu, D., Guo, L. and Dai, T. [2018], 'A LSTM Based Model for Personalized Context-Aware Citation Recommendation', *IEEE Access* **6**, 59618–59627.

Zequn Gao [2015], Examining influences of publication dates on citation recommendation systems, *in* '2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)', IEEE, pp. 1400–1405.

Zhang, C., Swami, A. and Chawla, N. V. [2018], 'CARL : Content-Aware Representation Learning for Heterogeneous Networks'.

Zhao, W., Wu, R. and Liu, H. [2016], 'Paper recommendation based on the knowledge gap between a researcher's background knowledge and research target', *Information Processing & Management* **52**(5), 976–988.

Zhou, Q., Chen, X. and Chen, C. [2014], Authoritative Scholarly Paper Recommendation Based on Paper Communities, *in* '2014 IEEE 17th International Conference on Computational Science and Engineering', IEEE, pp. 1536–1540.

Annexe A

Annexes

A.1 Transcription de la proposition initiale du CETIC

Source : `https://www.cetic.be/Systeme-de-recommandation-pour-les-references-bibliographiques`

Contexte

La publication scientifique fait l'objet d'une activité très importante et sans cesse croissante, à tel point qu'il est difficile aujourd'hui de suivre l'actualité de la recherche, ne serait-ce que sur une thématique précise. Une veille de la littérature doit cependant être réalisée par les chercheurs afin que ceux-ci puisse mener à bien un travail d'innovation.

Une problématique connexe à celle de la veille scientifique concerne la nécessité d'établir des relations entre le travail d'un chercheur et celui de ses collègues. En particulier, lors de la rédaction d'un article scientifique, l'usage veut qu'une partie de cet article soit consacrée à l'exposition d'un état de l'art dans lequel les travaux similaires à ceux présentés dans l'article sont présentés et discutés. Cela permet de situer l'article dans un contexte de recherche et de mettre en perspective la contribution des auteurs. Lorsque l'article couvre une thématique pour laquelle les auteurs ont peu d'expérience, il peut s'avérer difficile de mettre en évidence les publications pertinentes dans le cadre de leur travail.

Des travaux récents contribuent à la gestion de ces problèmes en proposant des outils analysant automatiquement des corpus de publications dans le but de présenter une information pertinente et utile aux chercheurs. Il est alors possible de procéder à une exploration ciblée de la littérature.

Travail à réaliser

Cette proposition de stage comportent deux objectifs complémentaires. De part leur ampleur, un stagiaire ne pourra, a priori, travailler que sur un seul de ceux-ci.

Le **premier objectif** consiste en l'installation au sein de l'infrastructure du Cetic d'une application de gestion des publications similaire à *Arxiv-Sanity*. Nous nous intéresserons principalement à son module de recommandations capable de sélectionner, parmi l'ensemble des publications récentes, les plus susceptibles d'intéresser un utilisateur. Après avoir réalisé un état de l'art des solutions existantes, le stagiaire déploiera l'une de ces solutions, de sorte qu'il soit possible de consulter des publications intéressantes dans une ou des thématique(s) donnée(s), ou encore de recevoir régulièrement par e-mail des recommandations de lecture.

Le **second objectif** [choisi dans le cadre de ce travail] consiste en la réalisation d'un système distribué capable de recommander aux auteurs rédigeant un article scientifique un ensemble de publications pertinentes sur base des publications déjà citées dans l'article rédigé. Le système devra extraire, à partir d'un corpus de publications, les informations pertinentes pour l'établissement d'un système de recommandation. Le système recommandera alors des publications à citer en temps réel. Le stage sera finalisé par la réalisation d'un prototype d'application Web offrant à l'utilisateur la possibilité de soumettre une ébauche d'article scientifique et d'obtenir en retour les publications recommandées par le système.

Encadrement

L'entièreté du travail sera encadré. Le ou la stagiaire utilisera une plateforme de développement permettant le suivi constant de ses progrès. Elle ou il devra également faire preuve d'autonomie et d'esprit critique lorsque des choix techniques et technologiques devront être opérés.

Contact : Mathieu Goeminne (`mathieu.goeminne@cetic.be`)

A.2 Questionnaire interview UNamur

1. Pouvez-vous vous présenter en quelques mots ? → *cerner le profil du chercheur (expérience, thématiques d'intérêt, production, etc.).*
2. Quelles sont les occasions qui vous poussent à réaliser une recherche bibliographique ? → *identifier les différents motivations/objectifs d'une recherche, notamment les délivrables attendus (articles, projets de recherche, état de l'art, thèse, etc.).*
3. Pour ces différentes occasions, est-ce que la recherche bibliographique doit être très précise ? faite rapidement ? ou bien avez-vous le temps ? → *découvrir les différents contextes dans lesquels une recherche est réalisée (urgence, contraintes, etc.).*
4. Comment procédez-vous lorsque vous réalisez une recherche bibliographique ? → *préciser le processus général de recherche, notamment les différentes étapes.*
5. Quels outils de recherche utilisez-vous ? En êtes-vous satisfait ? → *recenser les outils utilisés, avec un feedback lié à leur utilisation.*
6. De manière générale, quelles sont les principales difficultés auxquelles vous êtes confronté dans le cadre d'une recherche bibliographique ? → *identifier les principaux écueils (que le système à concevoir pourrait solutionner).*
7. Outre la qualité du contenu proprement dit, comment évaluez-vous qualitativement la production scientifique ? Quels critères formels utilisez-vous (auteurs, organes de publication, bibliographie, etc.) ?
8. Avez-vous déjà utilisé un système de recommandation de références bibliographiques (ou dans un autre domaine) ? Qu'en avez-vous pensé ? → *recueillir les éventuelles expériences avec un système de recommandation,* → *si l'utilisateur bloque, mentionner les recommandations de Google Scholar, ResearchGate, Academia, Mendeley, etc.,* → *creuser la question de la veille documentaire (comment procède la personne en général, est-ce qu'il utilise des recommandations).*
9. Si vous deviez disposer d'un système de recommandation dans le cadre de vos recherches, quelles fonctionnalités souhaiteriez-vous ? Comment verriez-vous cet outil ? De manière plus générale, quels sont les besoins/manques que cet outil pourrait combler dans vos recherches bibliographiques ? → *alimenter un pool d'idées quant aux besoins du système à concevoir (fonctionnalités, IHM, scénarios d'interaction, etc.).*

A.3 Tableau des méthodes

Voir fichier externe : `annexe_tableauMethodes.ods`

A.4 Résultats de l'étude utilisateur

Voir fichier externe : `annexe_reponsesEtudeUtilisateur.csv`

A.5 Code source de la comparaison des méthodes

Voir fichiers externes : `annexe_sourceEval.zip` et `annexe_datasetsEval.zip`

A.6 Code source du prototype

Voir fichier externe : `annexe_sourceProto.zip`

A.7 Vidéo de démonstration du prototype

Voir fichier externe : `annexe_videoProto.mkv`